Saarland University

Department of Computational Linguistics

Software Project: **Neural Network**

Predict the Winner in League of Legends using GANs (Web Application)

Author:
Koushik Chowdhury
Matriculation: 2572865

Supervisor:
Prof. Dr. Dietrich Klakow

7th April 2019

Abstract

League of Legends is one of the most popular online multiplayer video games in the world. It is a team-based online game. It has a large fan base in the world. Each team selects characters for their side. These characters are known as champions. These champions have certain abilities, some of them are weak and some of them are strong enough. A game result entirely depends on the strength ability of the champions and the members of the team. To know the strength abilities of team members is quite impossible, as this is an online multiplayer game, most of the people love to play with fake names. On the other hand, it is possible to know the strength and weakness of a champion. Therefore, picking the best champion plays an important role here. It increases the probability of a win for a team. To meet this fact, Neural Network (Generative adversarial networks) knowledge has been used in this project. This project aims at providing an interface where a team can pick the best probable champions. Interface suggests champions based on their winning probabilities. We employ neural network technology to predict the best probable champions with the highest probability of winning. In this report, we illustrate the development process of this interface.

Contents

1 Introduction

League of legends is developed and published by Riot Games for macOS and Windows user [1]. It has 27 million active users per day and over 100 million users per month. There are two teams in each game. Each team consists of five players. One team is the team that we select and another team is the opponent team. That means each team can select five champions. So, 10 champions can be selected in per game. Each player has his own champion. Champions are unique. There should be no overlaps of champions between two team. For example, if the green team picks the champion 'Alistar', then the white team will not able to select 'Alistar' for their side. Our goal is to predict the best champion with the help Neural Network (GANs). It will help a team to pick a champion that will give the highest chance of accuracy to win the game. In general, suppose, an opponent team selects all their champions. Now it's our turn to pick the five champions for our team. The system will suggest the champions ascendingly to us with the winning percentages. Our task is to pick the champions that give the highest winning rates. The whole project has four parts, such as Generator, Adversarial Classifier, Regressor and Web application. A generator predicts important in game statistics for the given champion combination. An adversarial classifier trains the generator to produce correct data, a regressor estimates the final winning chances by taking the predicted game information, Web application takes the final winning chances from the regressor and shows them graphically to the game users, so that they can pick the best champions for their teams. Our project is intended to take a large set of data from League of legends games and perform predictive analysis using generative adversarial networks. This report concerns the web application part. The report describes the whole software cycle that has been used to develop the web application. To develop the application, we follow a recognized software life cycle model. There are many software development life cycle models. We decided to follow the Waterfall model. This is one of the simplest software life cycle models. Nowadays it is not so popular model for developing large marketing application but for a small application like ours, this model is the most suitable one.

2 Background Study

2.1 Literature Review

Neural network is one of the most used techniques in statistical learning. Many researchers have shown the way and the purposes of neural network for different fields in their research. For our project, we have gone through some of the relevant research papers that recently published. Most of the research papers are about prediction, measuring performance,

forecasting or determining the future situation. One of the research paper has tried to determine the games outcomes in Dota 2 by using Artificial Neural Network [2]. Data 2 is also a popular online video game. This paper predicts the game outcomes based on dota 2 characters selection made in each game. Some researchers have applied neural network techniques to predict student performance in the faculty of engineering and information technology [3]. Medical data has now an interesting field for applying neural network techniques. By using neural network, it is possible to predict the specific disease and give probable treatment about that specific disease. Willem de Haan, Yolande AL Pijnenburg, Rob LM Strijers have shown a functional neural network analysis by using EEG and graph theory in frontotemporal dementia and Alzheimer's disease [4]. We have gathered information about how to classify data successfully and apply neural network techniques based on some features from these research papers. In our project, we have collected data set for predicting the winner of the league of legends based on some attributes.

2.2 Champions

Champions are recognized as game controller character in the league of legends [5]. Each champion is controlled by a player, a player who is a human being. Each champion has different levels of quality and ability. They can set the game behavior and environment. They have some strength such as summoner spells, runes, mastereries, etc. These things can boost up the strength of a champion. There are several classes of champions like controller, fighter, mage, specialist, slayer, etc. Therefore, based on the classes, a group of champions can act differently but in the end, all champion's target is the same, 'win the game'.

2.3 Dataset

One million data points have been downloaded from the league of legend's official data source [6]. The format of data [JSON] is as follows:


```
{
    "seasonId": 11,
    "gameId": 3887470339, …
    "teams": [{
        "win": "Win",
        "baronKills": 1, …
    }, {…}],
    "participants": [{
        "championId": 36, …
    }, …]
}
```


Figure 1. format of data [JSON]

So, there are one million JSON files. Each JSON files represents one data point of a single game. All the JSON files (1 million files) have been converted into one CSV file. After converted into a CSV file, we have furbished our data sets for analysis.

3 Methodology

This section describes the full process of development. As mentioned earlier, this application will help the league of legends user to pick the best champion for their team. Champion selection is one of the main challenges in the league of legends game. The final game result depends on the ability of the champion, also it is necessary that the team members should have experience in the league of legends, otherwise, picking the best champion will not help them.

3.1 Features of Application

The application has some features as follows:

- Add champions
- Delete champions
- Search champions
- Probable Winning rate available
- Team winning percentage based on picked champions
- Champions Details
- Instructions
- Videos

3.2 Waterfall Model

There are basically six levels in the waterfall model. There are, gathering the requirement, Design UI, Coding implementation, Testing, Deployment and maintenance. The main benefit of the waterfall model is that it gives departmentalization control. Each step depends on each other. If one fall, the full system would collapse. Diagrammatic representation of the Waterfall model is illustrated below.

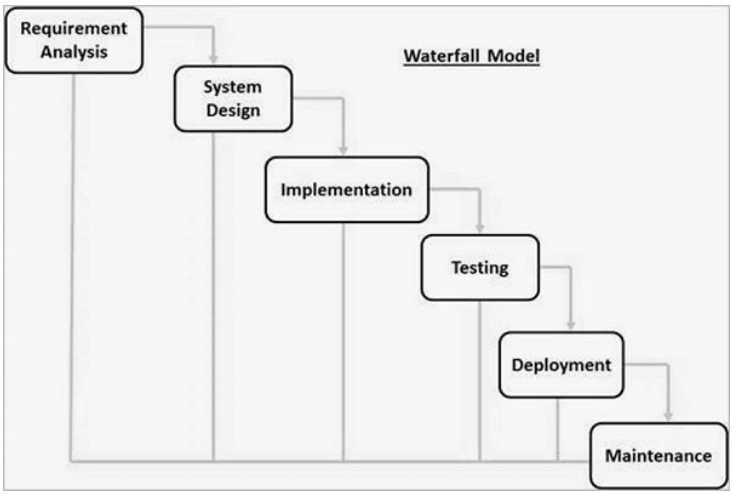

Figure 2. The Water Model (source: tutorialspoint)

There are some reasons for using this model. One can easily understand and use this model. If the requirements are fixed and clear, this life cycle works perfectly.

3.3 Gathering Requirement

Requirement gathering is a prior step for application development. It is important to know that what are the requirements would need to develop an application. The main requirement of this project is to get the champions winning probabilities from regressor. A JSON file is created with the champions id and their winning probabilities. The JSON file integrates with the application. The second most important requirement is to collect all champions ids, names, icons. Champions icons (images), names and ids are taken from the League of legend's official API [7]. There is also some system requirement for this project such as,

- The application should support on any web browser.
- The application should use a specifies server (Server Name: XAMPP).
- The application should be developed using JavaScript, Ajax, HTML, CSS.
- The application should test all the specified requirements using automated scripts (Framework: Protractor)
- The application should be tested manually by the help of real users of the League of Legends.

3.4 Design UI

Designing the interface is the 2nd step of the designing process. If a system has no good design, it will not attract the user. Our design includes HTML, CSS and some other web references [8][9]. Raw HTML and CSS have been written for the application, also external fonts and design references have been used to complete the Front end task. Two separate part has been designed for the application. Each part contains 143 champions with pick functionality. There are also two separate boxes. Each box has five columns. One column is equal to one champion. Whenever a user picks a champion, that champion will move to the box. Also, the interface has a remove functionality. If the user removes a champion, it will return back to the champion list part. The interface has a searching option. If a user wants, he can search his desire champion from the list and can pick that champion.

Figure 3. The UI Design

Whenever a user points his mouse on the top champion, it will display the champions details such as name, winning probable scores, etc. The interface has also other features such as videos where one can see the previous 'League of Legends World Championship' and also, one can able to stream the live world championship when it will happen. The interface has another feature named 'Details' where user can see the descriptive summary of a champion.

3.5 Coding Implementation

As a high-level programming language, JavaScript has been used. Also, Ajax has been used to send and retrieve informations from the server. Full JavaScript implementation has been broken into several modules such as JSON handler, add champion to the team, check for duplicate champions, delete champions from the team, Reorder based on champion's winning rate, change scores, search box, etc. At the end, when all the champions are picked, calculated winning rate will be shown to the display which has been implemented

by the JavaScript. Almost 2600 lines of code have been written including JavaScript, Ajax and HTML. There is also a separate CSS files. Also, there is some references style content added from the web [8][9]. A small snippet part of the code is given below.

```
// Add champion to team
$('body').on('click', '.teampicker-champion .btn-blue-f', function(e) {
    e.preventDefault();

    var team = $(this).closest('[data-team]').data('team');
    var champion = $(this).closest('.teampickerlist-champion').data('champion');
    var score = parseFloat($(this).prev().text());

    if (team == 'myteam') {
        team = ".teampicker-myteam .teampicker-champion";
    } else {
        team = ".teampicker-opponent .teampicker-champion";
    }
```

Figure 4. Small snippet part of add champion

This code represents the small portion of implementation from add champion part. Code binding part is one of the main challenges of this phase of the model. All the coding stuff are done on Sublime Text editor, which is a cross-platform open source editor.

3.6 Testing

Two most common methods of software testing are Black-Box and White-Box testing. In Black box testing, access of code is required. The user can do this testing. On the other hand, in white-box testing, the tester should know the code structure.

3.6.1 Black-Box Testing

Instead of doing the manual test by ourselves, we have taken feedback from the user. Because of the time limit, we were only able to do 2 test. Our testers are the real user of the legend of legends. They are the student of the UdS, Filip and Tajbeed. The results are drawn below.

Test name	Tester 1 (Filip)	Tester 2 (Tajbeed)
Functionality Error	1	1
Interface Error	1	3
Performance Error	2	1
Input Error	1	2
Integration Error	1	1

Figure 5. Feedback result (results out 5)

Five represents the highest number of errors and one represents the lowest number of errors.

3.6.2 White-Box Testing

This part of testing is done by us. We have used a popular testing tool named 'Protractor'. 'Protractor' is the end to end test framework. This framework is mainly used to test a web-based application. Also, because of the short time, it is not possible to go through every module for testing. A small snippet test result from 'add champions' module is given below.

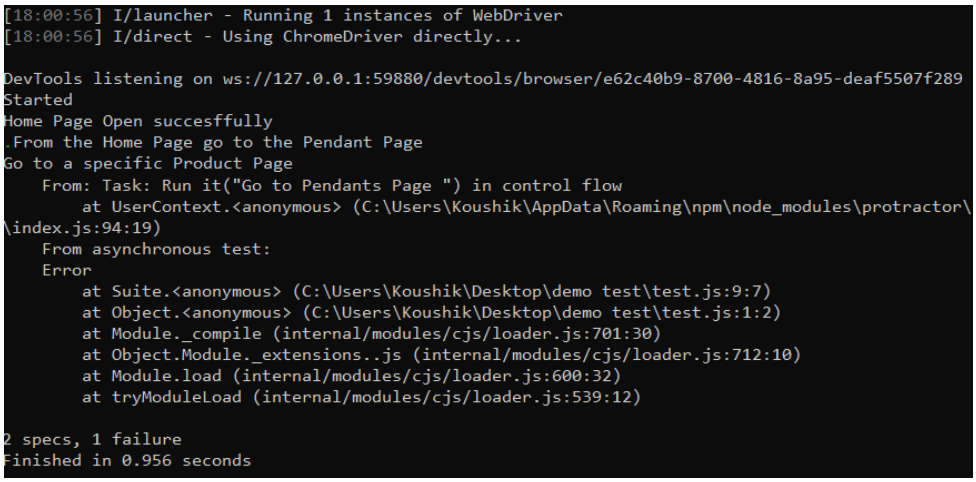

Figure 6. Result from Protractor for 'add champion'

3.7 Deployment and Maintenance

Only the 1st version is released for this web application. The full application is running on a local server named 'XAMPP', which is an open source cross-platform server for the web. The application is not yet uploaded in the online server because it is still on modifying process. Also, Budget and permission are needed from the league of legends authority if we wish to publish the application. This will take lots of time. If we want to upload the application, we need to go through some legal discussion.

3.8 Limitations

There is some limitation of this project. At the beginning of the project, some of the team members had no previous experience of any such online battle games. So, it has taken much time to understand the game. Also, as mentioned above, the site is running

on a localhost and it is not possible to access the site from other computers. The winning accuracy is static for the champions. This has to be dynamic for each selection of a champion. This web application is not well-suited for marketing but with modification, time and budget, it will be acceptable for marketing.

3.9 Updated Version

In version 2.0, the design of the interface will be modified. A specific database will be used to store the data. 'Summoner' spells will be implemented in the application. The system will suggest the best 'Summoner' spell to the player based on the champion's ability. With 'summoner' spell, the result will be more accurate, also the application will be more reliable to the league of the legends users.

4 Conclusion

This project has brought an interesting technique for predicting the winner for league of legends using Neural Network. In this project, a concept has been illustrated and analysis has been done by generative adversarial networks. Real live data has been collected for this project. Then generator, adversarial classifiers regressor have been done and after that, a web application has been created. Nowadays, online games such as the League of Legends, Dota 2, etc, become more popular to internet users as well as these online games achieved a fantastic business status. Our belief is that this application will encourage the online game users to play this battle game as now they can know their winning chances from this application.

References

[1] The League of Legends
https://na.leagueoflegends.com/en/game-info/get-started/what-is-lol/

[2] Widin, Viktor, and Julien Adler. "On using Artificial Neural Network models to predict game outcomes in Dota 2." (2017).

[3] Naser, Samy Abu, et al. "Predicting student performance using artificial neural network: In the faculty of engineering and information technology." International Journal of Hybrid Information Technology 8.2 (2015): 221-228.

[4] de Haan, Willem, et al. "Functional neural network analysis in frontotemporal dementia and Alzheimer's disease using EEG and graph theory." BMC neuroscience 10.1 (2009): 101.

[5] The League of Legends: Champions
https://leagueoflegends.fandom.com/wiki/Champion

[6] Riot Games API
https://developer.riotgames.com/

[7] League of Legends API
https://ddragon.leagueoflegends.com/cdn/8.24.1/data/en_US/champion.json

[8] Style References 1
https://fonts.googleapis.com/css?family=Lato:400,700|Exo:600

[9] Style References 2
https://cdnjs.cloudflare.com/ajax/libs/normalize/3.0.2/normalize.min.css