Predicting Multiplayer Online Battle Arena (MOBA) Game Outcome Based on Hero Draft Data

MSc Research Project
Data Analytics

Weiqi Wang
x15021033

School of Computing
National College of Ireland

Supervisor: Dr. Ralf Bierig

National College of Ireland
Project Submission Sheet – 2015/2016
School of Computing

Student Name:	Weiqi Wang
Student ID:	x15021033
Programme:	Data Analytics
Year:	2016
Module:	MSc Research Project
Lecturer:	Dr. Ralf Bierig
Submission Due Date:	21/12/2016
Project Title:	Predicting Multiplayer Online Battle Arena (MOBA) Game Outcome Based on Hero Draft Data
Word Count:	XXX

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	
Date:	21st December 2016

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
3. Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Predicting Multiplayer Online Battle Arena (MOBA) Game Outcome Based on Hero Draft Data

Weiqi Wang
x15021033
MSc Research Project in Data Analytics

21st December 2016

Abstract

DotA 2 is a popular multi-player online battle area (MOBA) game. A critical part of the game play involves choosing from a pool of more than one hundred heroes to form two five-players team. However, as different heroes have their unique attributes and skill sets, selecting a strong combination of heroes (i.e., hero drafting) is an challenging task for new players which requires extensive knowledge and experience. Previous studies have shown that using hero draft data alone can achieve as high as 69.8% of accuracy in predicting game outcomes. However, many aspects in hero draft remains to be further investigated. In this study, we aimed to achieve higher accuracy by adding game length as an input feature. In addition, we used multi-layer feedforward neural networks to predict the game outcome with GPU enabled. However, the results showed that adding game length does not improve the performance significantly nor did neural networks outperform logistic regression significantly.

1 Introduction

1.1 Background

DotA (short for "Defense of the Ancients") is a multiplayer online battle arena (MOBA) game originally developed and maintained by some enthusiasts via the Warcraft III platform. It has gained a great popularity over the time. Seeing great potential of the game, Valve hired its main developer, Icefrog, to develop its second generation, DotA 2, via the Steam platform. Since its official release in 2013, DotA 2 has become a hit. Currently, there are around 90,713,551[1] players who own this game in their Steam account. In March 2016, DotA 2 researched its all-time peak of online players, 1,295,5114[2]. At the time of writing this report, the median playtime in the last two weeks was 10.3 hours[3]. The popularity of the game is not only indicated by the magnitude of player community and enthusiasm of the players, but is also indicated by the development of it being a

[1]Data: https://steamdb.info/app/570/graphs/
[2]Ditto.
[3]Ditto.

professional sport. For example, the most recent major championship, The International 2016, attracted 16 professional teams to compete for a prize pool of $20,770,460 USD[4].

1.2 Game Mechanics

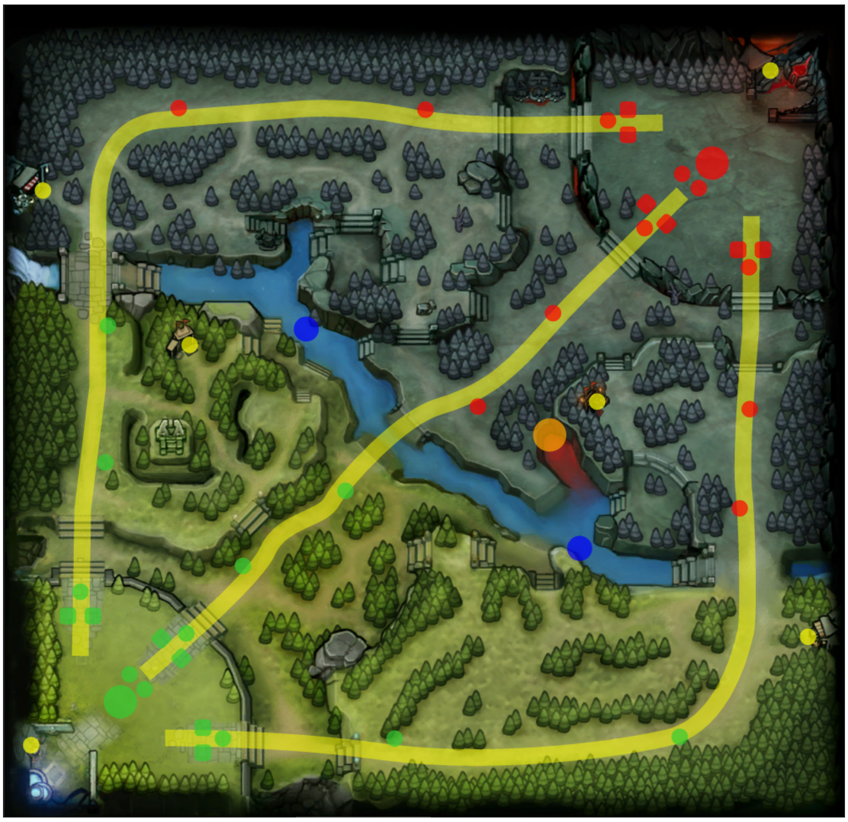

Figure 1: Map of DotA 2

The most basic form of the game is a 5 vs. 5 match. In each match, ten players form into two teams. To win a match, one side have to destroy the main building, namely "Ancient" (the two largest dots in the map above), of the other side. One side of the main building is located in the bottom left of the map, while the other side of main building is located in the upper right of the map. The map is roughly symmetric.

Each player can choose one hero from a pool of more than one hundred heroes. Each hero has a unique skill set (usually consisting four unique skills). Based on the character of their skills, heroes play different roles in a team. For example, some heroes are responsible for dealing massive damage to the enemy team while other heroes are responsible for healing their allies. In addition to their basic roles in the game, heroes have either synergistic or antagonistic relationships with other heroes as a result of their basic attributes (e.g., melee or range attack) and skill set. This is true for heroes both in one team and in counter team. No hero is almighty–every hero has some heroes that counters it. Therefore, picking the right combination of heroes (i.e., hero draft) is considered a critical aspect of the game play. In professional matches, hero draft is usually done by team leaders. Indeed, in a recent research, Summerville et al. (2016) studied the possibility of automating this hero draft process.

[4]Ditto.

1.3 Research Questions

For both professional and amateur players, improving their skills and understanding of the game is critical for those who aims to play better. Therefore, many players have studied the game's mechanics extensively. However, as mentioned above, choosing the right combinations of hero is an extremely challenging yet critical task. In fact, there are $C(113, 10) = \frac{113!}{(10!(113-10)!)} = 62088566355816$ possible hero combinations. Although in practice, some combinations are more commonly used than others. It is never the less a very difficult task considering that one need to be familiar with the heroes and the right combinations. Therefore, it is always a very unpleasant experience for new players to learn how to choose heroes. In addition, if they pick the hero that are does not fit with the rest of the team. Other players may express strong disappointment towards these new players. In deed, Semenov et al. (2016) found that for matches in the lower skill band, it is easier to predict the game outcome based on hero draft. They articulated that this is because new players tend to pick imbalanced hero combinations. Therefore, the results are more predictable as one side suffers from greater disadvantage from bad hero draft. To address this problem, many third-party websites and products are developed to help players master hero draft. For example, becomethegamer.com developed a plugin that helps players choose hero in game.

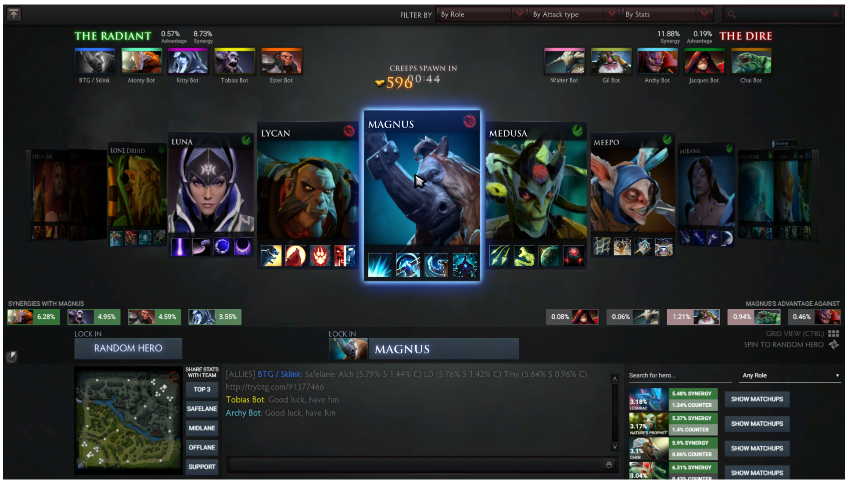

Figure 2: Drafting Screen with becomethegamer's Plugin

Other than the functionalities seen in this product, a possible decision support system could also display the wining probability based on hero draft. This is essentially a classification problem in machine learning. In addition, the data generated by players playing this game is enormous. These data in turn could be used to build the model. Furthermore, using the model for predicting the game outcome, one could develop hero draft recommender system to help players make better decision during the draft stage. Although DotA 2 has been released for couple years now, there hasn't been many studies of the game. This is partly due to the fact that the game's API was only released in 2012 and is still in beta test now. Users have reported that the API breaks from time to time, which makes it even harder for researchers to collect data consistently. None the less, the Steam API allows users to collect large volume of data in very fine detail. For example, match detail data include individual players' gold per minute, experience per minute, kills and deaths and so on.

Researchers have used different methodologies to study the game and its related issues. For example, the earliest study (Nuangjumnonga et al.; 2012) investigated the relationship between different play styles and leadership styles using data collected with closed-end surveys. Some studies (Drachen, Yancey, Maguire, Chu, Wang, Mahlmann, Schubert and Klabajan; 2014) parsed the game replay data and studied the topological patterns of players. Other studies (Conley and Perry; 2013) tried to predict the outcome of the match based on different data.

A major area of research is to predict the match outcome based on hero drafts data. As discussed above, this could be potentially very helpful for new players. The study that initiated this line of research was conducted by Conley and Perry (2013). Using simple algorithms such as Logistic Regression and k-Nearest-Neighbourhood (kNN), they were able to achieve an accuracy of 69.8% in predicting the match outcome in the test set. Inspired by their study, many other researchers further explored this topic. For example, some studies (Kalyanaraman; 2015) used more advanced algorithms such as Genetic Algorithm or Random Forest. While other studies (Kinkade and Lim; n.d.) tried to model the relationship between hero pairs more explicitly and found that this could greatly improve the performance. However, some problems still need to be further investigated. For example, neural networks were not considered by previous studies. Maybe this is because people suspect that it might take too long to train the model. However, recent advancement of deep learning that exploits GPU has made it possible to train a neural network much faster than before. Another possibility is to use game length as an input feature. An inspection of individual heroes' win rates over time suggest that heroes' win rates are not constant. Instead, hero win rates changes as the game length changes. For example, in the plot below, the hero's win rates gradually reaches its peak around 40 minutes, then starts to drop. Similarly, win rates of a particular combination of five heroes may also changes over time. A problem is that game length is a data obtained after the game finishes. There is no way to use it as an input at the beginning of the game. Despite that we cannot use the exact game duration as input, we could use time intervals as an approximation. For example, win rates of game length within 15 minutes and more than an hour could be different. In this way, the decision support system could display winning probability for different time intervals. This could also help players form different strategies.

In sum, the aim of this study is to further explore the potential of using hero draft data to predict match outcome in DotA 2. More specifically, this study tries to answer the following research questions:

1. Does adding game length as an input feature improve a particular model's performance?

2. Does neural network perform better than other algorithms?

This report is structured in the following manner: the next subsection familiarizes you with the basic concepts of neural networks; the second part of this report briefly reviews previous studies and offer a context for the current research; the third part of this report describes and explores the dataset in detail to gain a better understanding of the dataset; the fourth part of this report describes the implementation details; finally, an evaluation of the experiments' results is offered and a conclusion is drawn.

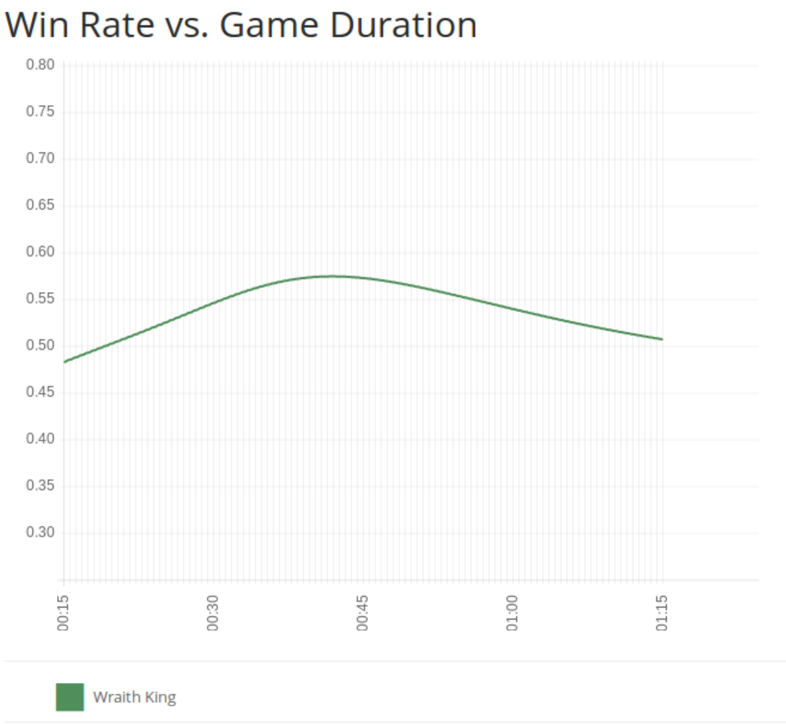

Figure 3: An Example of Individual Hero's Win Rate vs. Duration

1.4 Neural network

The emergence of neural network was inspired by the biological structures of human brain (Demuth et al.; 2014). Like the human brain, a standard neural network is consisted of many neurons (Schmidhuber; 2015). These neurons are connected to other neurons and form into different layers. By receiving data from the previous layer, neurons can be activated and pass data to the next layer. Neural networks have many different variants both in terms of their architecture design and hyper-parameter configurations. The most basic type of neural network models are called deep feedforward networks (Goodfellow et al.; 2016, p. 168). Deep feedfoward networks are also referred to as feedforward neural networks or multi-layer perceptrons (MLPs). In feedforward neural networks, information flows from the input to the output. The output layers cannot give feedback to the previous layer. Therefore, it is called feedforward neural network (Goodfellow et al.; 2016, p. 168). A feedforward neural network consists three types of layers, that is, an input layers as the first layer, one or more hidden layers, and an output layer as the last layer.

2 Related Work

2.1 Initial Studies

Game analytics as a means of helping game companies make more informed decisions has been proven very successful practice (El-Nasr et al.; 2013). However, the study of DotA 2 only began recently. The first study on DotA 2 was done by Nuangjumnonga, Mitomo et al. (2012). Initial studies investigated different aspects of the game using different methodologies. For example, Nuangjumnonga et al. (2012) examined the relationship between leadership styles and game roles using data collected from surveys. In another study, Pobiedina et al. (2013) statistically tested several hypotheses regarding the impact of team formation on the game outcome. Some other studies analysed the spatio-temporal patterns of game play using parsed game replay data (Drachen et al.; 2014; Yang et al.; 2014; Rioult et al.; 2014).

2.2 Game Outcome Prediction

Although these studies greatly enhanced our understanding of the game and its connections with our everyday life. However, a major line of research now is to predict game outcomes based on certain information such as hero draft. This line of research was initiated by Conley and Perry (2013). Since DotA 2 has more than one hundred unique heroes playing different roles in the game, hero drafting becomes a major aspect of the game. For example, pick and ban phase (both sides pick and ban heroes) in the professional matches is often considered one of the most challenging task which requires making decisions strategically. This task is often conducted by the team leader. Therefore, predicting the game outcome using hero draft data becomes a very valuable topic. A system could be implemented to help new players to learn about hero selection. Professional teams could also be benefited from a similar decision support system in their daily training. Using logistic regression, Conley and Perry (2013) achieved 69.8% of accuracy at a number of 18,000 training set. However, since logistic regression is a linear classifier and unable to capture the "synergistic and antagonistic relationships between heroes", they further used k-Nearest Nighbours to predict the game outcome. With the kNN model, they achieved 67.43% accuracy. However, a major drawback of using this model is that it takes a long time to train the model. It took them more than 12 hours to train their model using cross-validation.

Although logistic regression achieved higher accuracy for predicting the game outcome based on hero draft, logistic regression could not take the relationship between heroes into account. For example, some heroes perform much better in one particular team setting while other heroes may not have such advantage. On the other hand, most heroes counters some other heroes in the enemy team and also are countered by other heroes. Therefore, it is interesting to investigate the effect of heroes combinations. Based on Conley and Perry (2013) work, Agarwala and Pearce (2014) advance their methodology by incorporating the relationships between heroes. To cluster the roles of different heroes, and model their interactions, they used PCA analysis of the heroes' performance statistics (kills, deaths, gold per minute etc.). However, their results were not as good as the results in (Conley and Perry; 2013) study. Their new model only achieved 57% accuracy while their model without hero interactions achieved 62% accuracy.

In another study, Yang et al. (2014) manually added 50 two-hero combinations as new input features. However their results were not as good as Conley and Perry (2013) either. Kalyanaraman (2015) first used hero roles as one of the input feature implicitly. Using an ensemble combining Genetic Algorithm and Logistic Regression, their model achieved 74.1% accuracy. In a similar study, Kinkade and Lim (n.d.) got 72.9% accuracy using logistic regression by adding two hero win rates.

Most recently, Semenov et al. (2016) used xgBoost and Factor Machine to predict the game results. In addition, they used some other metrics for evaluation other than accuracy to assess their models. Therefore, it is difficult to compare the results of their study to the previous studies.

Although some previous studies achieved relatively high accuracy, only a very limited types of models were used, such as logistic regression, random forest and so on. No previous studies have explored neural networks. Since it is perfect classifier to model nonlinear relationship and interactions, in this study, we will explore the performance of neural networks on this task. In addition, as discussed in the Introduction, the win rate of individual heroes as well as hero combinations are not temporally static. Rather, win rate are dynamic. In other words, there is a time window for a particular hero of hero combination to achieve its optimal potential. If players do not finish the game within that time window, their win rate could drop drastically. This timing feature was not explored in previous studies, therefore, in the present, we aim to test the performance by adding this game length feature.

3 Methodology

This section describes the methodology procedures of the current study, including:

- data transformation and pre-processing
- descriptive and exploratory data analysis
- data modelling
- model evaluation

3.1 Data Pre-processing

The dataset used in the current study is provided by Semenov et al. (2016). According to Semenov et al. (2016), the dataset was collected via Steam API from 11/02/2016 and 02/03/2016. A total number of 5,071,858 match records is in the dataset. The original dataset has 15 features. However, not all the features are used in modelling, only duration and hero draft related features are used in modelling. In addition, the hero drafts data are converted into binary form.

The distribution of the original dataset is as follows:

	Captains mode	Random Draft	Ranked All Pick
Normal skill	33037	86472	2937087
High skill	5599	15560	917001
Very high skill	8840	39407	1028855

Not all records are used in the analysis. Data are filtered based on the several criteria which are also used by previous studies (e.g., (Conley and Perry; 2013)):

- Sometimes players leave the game before it finishes or are away from the keyboard (AFK) due to various reasons. Matches that have leavers or equivalent situations are excluded from analysis as these records may bias the dataset and thus should be discarded.
- Only those matches in the very high skill bracket are included. This is because highly skilled players' performance could represent the heroes true potential.
- All matches are at least 15 minutes long. Short games are potentially more biased due to various reasons. Therefore, matches shorter than 15 minutes are excluded.

3.1.1 Feature vector

The filtered dataset contains 912,501 records. Those columns containing hero ids are then binarized.

In the dataset used in the present study, there are a total number of 113 heroes. Therefore, the heroes are represented in the format below.

$$x_i = \begin{cases} 1, \text{if this hero is in the Radiant's team} \\ 0, \text{otherwise} \end{cases}$$

$$x_{i+113} = \begin{cases} 1, \text{if this hero is in the Dire's team} \\ 0, \text{otherwise} \end{cases}$$

The duration (in minutes) of the match is divided into the following intervals. Although the density map of the duration indicated different patterns of duration. Classifying duration in the following way is more intuitive and could be implemented in a decision support system.

$$x_{duration} = \begin{cases} 1, x \leq 15 \\ 2, 15 < x \leq 30 \\ 3, 30 < x \leq 45 \\ 4, 45 < x \leq 60 \\ 5, x > 60 \end{cases}$$

Finally, the outcome of the game is represented as the vector below.

$$y = \begin{cases} 1, \text{if Radiant won} \\ 0, \text{otherwise} \end{cases}$$

3.2 Descriptive & Exploratory Analysis

To explore the distribution of game length, we plotted the duration of game in minutes in a density plot.

To explore individual hero's capability, we ranked heroes' win rate (win rate = number of matches won with the presence of a particular hero / total number of matches with the presence of a particular hero). It can been seen from the plot, hero win rates vary greatly among heroes. The highest win rate is as high as 60% while the lowest one is only around 40%.

4 Implementation

4.1 Logistic regression

In the initial study conducted by Conley and Perry (2013), an accuracy of 69.8% was obtained on the test set using logistic regression at a 18,000 training set size. Although more precisely speaking, they used regularized logistic regression. They used Python's sci-kit learn library to perform logistic regression, which has some default parameters including L2 regularization. In the current study, we used also the sci-kit library for

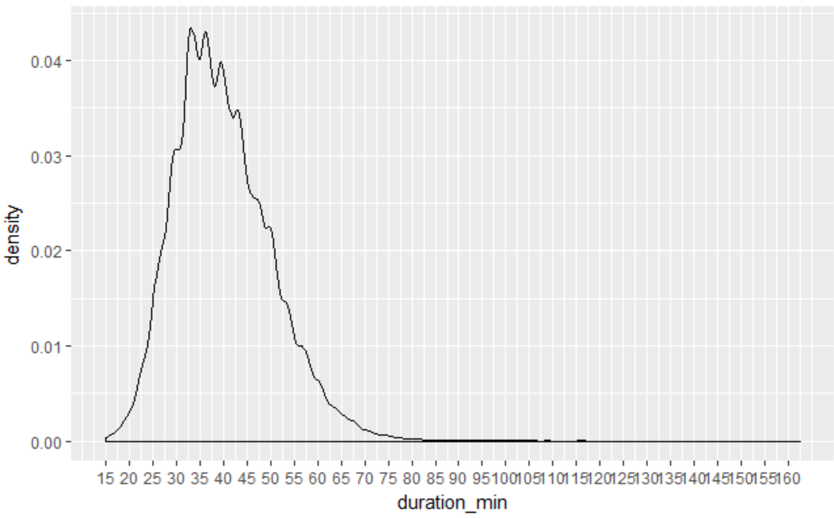

Figure 4: Density Plot of Game Duration (minutes)

logistic regression to get a baseline score on the dataset currently using. Additionally, a learning curve was plotted to identify the optimal training set size. As can be seen from the plot, the performance of logistic regression on the test set reaches plateau at around 200,000 training set size. However, the accuracy is much lower than the accuracy obtained by Conley and Perry (2013). Without the duration feature, a logistic regression with 5-fold cross validation produced an accuracy of 60.38%. When duration is added to the model, the accuracy increased to 61.04% with a minor improvement.

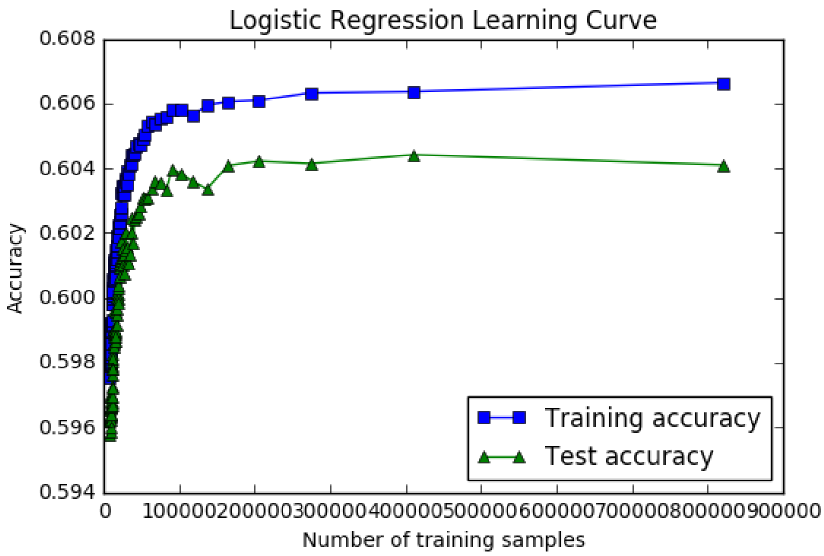

Figure 5: Logistic Regression's Learning Curve

4.2 Multi-Layer Feedforward Neural Network

Designing a neural network involves many decisions (Goodfellow et al.; 2016). First of all, we need to determine what optimizer to use, the metrics for evaluations. The input layer is determined by the data while the structure of the output layer depends on the task. Then, we need to design the architecture of the network. That is, the number of hidden layers and the number of neurons in each hidden layer. In addition, we need to decide what activation function to use for computing the values. Finally, we need to go through the process of hyper-parameters optimization such as learning rate, momentum, regularization parameters and so on.

A major advantage of the dataset used in the current project is its enormous size. Even after being preprocessed and filtered from the original dataset, there are still a number of 911,468 records in the subset. However, as seen in the learning curve of the logistic regression, the performance saturates at some point even more training data are fed into the classifier. This could also be the case for neural network. As can be seen from the learning curve below, the best performance occurs when the size of the training set is 70,000. Beyond that point, the performance of the neural network stops improving as the number of training set increases. Therefore, a number of 70,000 training set is used for training neural network.

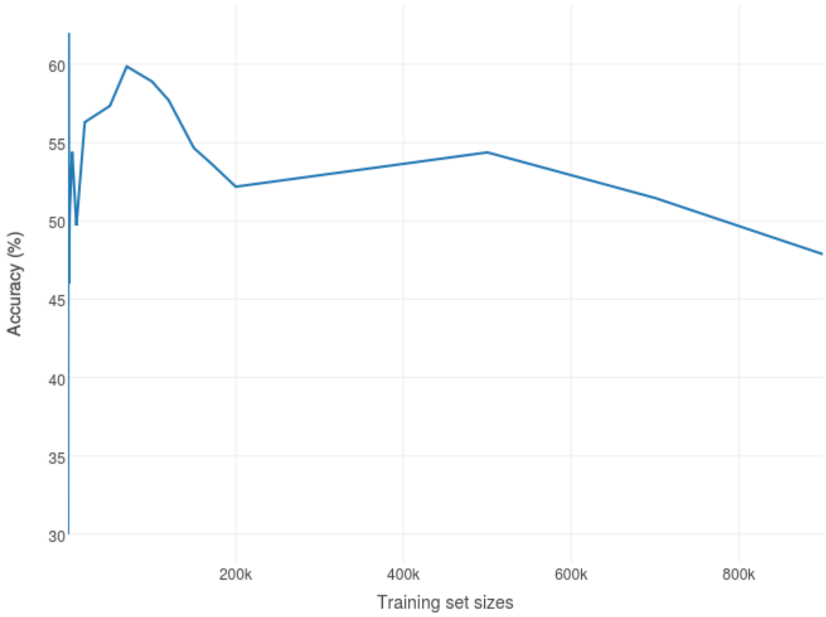

Figure 6: Test Accuracy vs. Training Set size

4.2.1 Optimizer

Unlike linear classifiers, the nonlinearity of neural networks mandates that they are trained with gradient-based methods iteratively. For example, stochastic Gradient Descent (SGD) is one of the most commonly used optimizers and is the default optimizer in many Deep Learning libraries (e.g., MXNet). In practice, instead of calculating with each records, a small batch size is chosen to calculate the gradient and update the the weights. Some other commonly used optimizers are RMSprop, Adagrad etc. In the current study, we experimented five optimizers (i.e., SGD, RMSprop, Adam, Adagrad, and Adadelta) using Bayesian optimization approach. The results showed that SGD yields the best performance. Therefore, the SGD optimizer is used.

4.2.2 Architecture design

For architectural configuration, a practitioner needs to choose the number of hidden layers, the number of neurons in each hidden layer, as well as the type of the neural network (e.g., feedforward nerual network, convolutional neural network, or recurrent neural network etc.).

There are many different kinds of the neural network architecture. Some are specialized in certain tasks. Convolution neural network (CNN) have been used extensively used in computer vision and related areas. It works very well in classifying images. Recurrent neural network (RNN) and its variants (e.g., Long Short Term Memory (LSTM)) are targeted to data with temporal features. It have been widely used in speech recognition and natural language processing. Due to the nature of the data in the current project, we are limited to use only feedforward neural networks.

Different structures of the feedforward neural network were experimented. We experimented with two hidden layers, three hidden layers, and four hidden layers with different number of nodes respectively. The three hidden layer structure performs the best.

4.2.3 Hyper-parameters optimization

In terms of tuning hyper-parameters of the neural network, it is essentially a hyper-parameters optimization problem. Different hyper-parameters form a search space. The goal is then to find the optimized combination of different hyper-parameters from the search space. An example of the search space is as follows:

$$\begin{aligned}
\text{learning rate} &= \{0.0001, 0.001, 0.01, 0.1\} \\
\text{momentum} &= \{0.5, 0.6, 0.7, 0.8\} \\
\text{drop out rate} &= \{0.1, 0.3, 0.5, 0.7\} \\
\text{batch size} &= \{10, 30, 50, 70, 100\}
\end{aligned}$$

These parameters forms a Cartesian product search space, which is very large. There are two basic techniques to find the best combinations. First, one could manually experiment with different hyper-parameters and look for patterns in the changes. Then, based on the findings, one could iteratively tuning different hyper-parameters and find a solution. The drawback of this method is obvious. It is very time consuming and requires a lot of manual work. The second basic approach is to conduct a grid search of the hyper-parameters. The idea is to exhaustively experiment with as much combinations as possible. Again, this is very time consuming and requires a lot of computational power. To deal with the defects of these two techniques, researchers have proposed different methods to boost and facilitate the procedure of hyper-parameters optimizations. For example, Bergstra and Bengio (2012) proposed a random search algorithm for hyper-parameters optimization. Another very popular method is referred to as Bayesian hyper-parameter optimization (Snoek et al.; 2012).

In the current study, different approaches are adopted. First, manual search was conducted to experiment with different configurations of layers and neurons. It was found that the model performs best with a three layer configuration. It is most probably that gradient vanishing or exploding has caused this problem.

Two deep learning libraries in the study, Keras and MXNet. MXNet (Chen et al.; 2015) is an efficient and flexible deep learning library and has gained a lot of attention recently. It is the only deep learning library that supports R with GPU computing. There are two libraries that supports Bayesian approach optimization in R and Python respectively. rBayesianOptimization is used to optimize parameters of the neural network implementation in MXNet. For Python, Hyperas is used for Bayesian optimization.

4.2.4 Epochs and batch size

An epoch is a complete iteration on the training set. Batch size refers to the number of records used to calculate gradient. It can be seen from the plot below that, as the number of epochs increases, the model becomes more and more overfitting on the training set. On the other hand, the performance on the test set reaches saturation in early epochs.

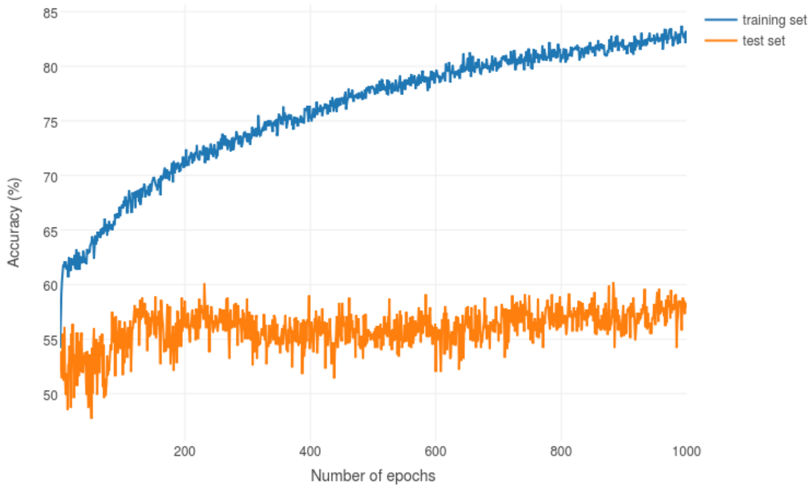

Figure 7: Overfitting

5 Evaluation

The performance of the multi-layer feedforward neural network did not achieve very high overall. The best results were similar to those produced by logistic regression. Although adding the duration of the game to the logistic regression resulted a minor improvement.

Table 1: Test Accuracy

	LR	LR (with duration)	MLP
Test accuracy (%)	60.38	61.04	0.588

Table 2: Confusion Matrix of MLP Results

	Acutal	
Prediction	0	1
0	1891	1667
1	1486	1956

6 Conclusion and Future Work

Hero drafting is a critical process of the DotA 2 game. However, due to the complexity of this process, new players often make bad decisions when choosing heroes. In order to help new players to choose hero, previous studies (e.g., (Conley and Perry; 2013)) have used hero draft data to predict the outcome of the game and recommend heroes based on the probabilities. However, previous did not take the advantage of the game length data. In this study, we used game length as an input feature to compare with the base line model. The results indicates that adding game length intervals generated a minor improvement in the performance. In addition, this study experimented with neural networks which no previous studies have experimented before. However, the results showed that neural network did not outperform logistic regression. In addition, the training process is much more demanding. However, it could also because an optimal set of parameters were not found in the current study. Future studies could add more features into the model such as the synergistic relationships between heroes as input features. In addition, future studies could use the kill ratios of the match as an output.

Acknowledgements

I would like to express my gratitude to my supervisor, Dr. Ralf Bierig for his guidance and help. I would also like to say thank you to Dr. Simon Caton for his invaluable suggestions.

References

Agarwala, A. and Pearce, M. (2014). Learning dota 2 team compositions.

Bergstra, J. and Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization, *Journal of Machine Learning Research* **13**(Feb): 281–305.
URL: *http://www.jmlr.org/papers/v13/bergstra12a.html*

Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., Xiao, T., Xu, B., Zhang, C. and Zhang, Z. (2015). Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems, *arXiv preprint arXiv:1512.01274* .

Conley, K. and Perry, D. (2013). How does he saw me? a recommendation engine for picking heroes in dota 2, *Np, nd Web* **7**.

Demuth, H. B., Beale, M. H., De Jess, O. and Hagan, M. T. (2014). *Neural Network Design*, 2nd edn, Martin Hagan, USA.

Drachen, A., Yancey, M., Maguire, J., Chu, D., Wang, I. Y., Mahlmann, T., Schubert, M. and Klabajan, D. (2014). Skill-based differences in spatio-temporal team behaviour in defence of the ancients 2 (dota 2), *Games Media Entertainment (GEM), 2014 IEEE*, pp. 1–8.

El-Nasr, M. S., Drachen, A. and Canossa, A. (2013). *Game Analytics: Maximizing the Value of Player Data*, Springer Publishing Company, Incorporated.

Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep Learning*, The MIT Press, Cambridge, MA.

Kalyanaraman, K. (2015). To win or not to win? a prediction model to determine the outcome of a dota2 match.

Kinkade, N. and Lim, K. y. K. (n.d.). Dota 2 win prediction.

Nuangjumnonga, T., Mitomo, H. et al. (2012). Leadership development through online gaming, *19th ITS Biennial Conference, Bangkok 2012: Moving Forward with Future Technologies-Opening a Platform for All*, number 72527, International Telecommunications Society (ITS).

Pobiedina, N., Neidhardt, J., d. C. Calatrava Moreno, M., Grad-Gyenge, L. and Werthner, H. (2013). On successful team formation: Statistical analysis of a multiplayer online game, *2013 IEEE 15th Conference on Business Informatics*, pp. 55–62.

Rioult, F., Mtivier, J.-P., Helleu, B., Scelles, N. and Durand, C. (2014). Mining tracks of competitive video games, *AASRI Procedia* **8**: 82 – 87.
URL: *http://www.sciencedirect.com/science/article/pii/S221267161400081X*

Schmidhuber, J. (2015). Deep learning in neural networks: An overview, *Neural Networks* **61**: 85 – 117.
URL: *http://www.sciencedirect.com/science/article/pii/S0893608014002135*

Semenov, A., Romov, P., Korolev, S., Yashkov, D. and Neklyudov, K. (2016). Performance of Machine Learning Algorithms in Predicting Game Outcome from Drafts in Dota 2, *Analysis of Images, Social Networks and Texts 5th International Conference, AIST 2016*, Springer International Publishing.

Snoek, J., Larochelle, H. and Adams, R. P. (2012). Practical Bayesian Optimization of Machine Learning Algorithms, *in* F. Pereira, C. J. C. Burges, L. Bottou and K. Q. Weinberger (eds), *Advances in Neural Information Processing Systems 25*, Curran Associates, Inc., pp. 2951–2959.
URL: *http://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf*

Summerville, A., Cook, M. and Steenhuisen, B. (2016). Draft-analysis of the ancients: Predicting draft picks in dota 2 using machine learning, *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Yang, P., Harrison, B. and Roberts, D. L. (2014). Identifying patterns in combat that are predictive of success in moba games, *Proceedings of Foundations of Digital Games* .