Deep leaning applied to drafting heroes in Dota 2

Isak Nyberg
ID1214 Artificial Intelligence and Applied Methods

12th January 2021

Abstract

In this project, we use Neural Networks and Deep Learning to predict the winning team of a Dota 2 match only based on information available at the draft. Match samples were collected from the OpenDota API and trained using a neural network with the match victors as the label of the samples. A model trained with just under 4 million samples from 2 million matches was able to label 61% of unseen samples correctly. This result does lie within the expected value for the importance of the draft. However, it is not significant enough to solely rely on for a real-world application, such as drafting in a competitive Dota 2 tournament. Instead the model is sufficient as an assistant to give suggestions for players rather.

1 Introduction

Dota 2 is a so called Multiplayer online battle arena game (MOBA) developed by Valve Software. This type of videogame has two teams of 5 play against each other with the use of characters with abilities, in the case of Dota 2, the characters that each player uses is called a hero. The ultimate goal of the game is for one team to collect enough resources to be able to attack and destroy the enemy base while also defending their own base from the attacks of the enemy. While the game is mostly focused on the player versus player combat another very decisive part of the game is which heroes that are picked by each team. Due to the different abilities that the different heroes possess, their likelihood for either team to win can drastically differ depending on which heroes are picked by either team, this is known as the draft. This concept is what will be explored in this project.

1.1 Background and Related work

Dota 2 has already been explored with the use of artificial intelligence (AI). The company OpenAI cofounded among others, Elon Musk and Sam Altman in 2015 has already deep learning to the game of Dota 2. Their project began in 2017 at the Dota 2 tournament The International 7 which is an annual Dota 2 tournament with multimillion-dollar price pools. At this event the AI made by the OpenAI team was able to beat one of the best human players in a one versus one match. This was the first time this had occurred. [1] Two year later in 2019 OpenAI showed up at The International 9 where they managed to beat the first-place team in a 5 vs 5 match. Solidifying the claim that AI was superior to humans. [2]

While OpenAI primarily focused on the gameplay, having drafts either being done by humans or predetermined. For this project an understanding of how the drafting in Dota 2 is required. The game has a few different picking modes but the primary mode that is the one used in Captains mode. The reason this will be used is because it is the most straight-forward form of drafting. The match begins with a captain from each team being presented with a screen of all 120 different heroes. Each captain take turn banning two heroes each, this means that the banned hero cannot be picked this match. Thereafter the picking phase begins where the captains alternate on picking 2 heroes each, a hero can never be picked twice by either team, and each pick is revealed to the enemy team. Thereafter another ban phase starts this time with 3 bans for each captain. Then another pick phase identical to the first one, followed by another ban phase of 2 heroes. Lastly each team get one final pick each. It is therefore strategic to save the most important hero for last since that way the opponent does not see what hero was picked and therefore cannot pick use that information when the pick a hero themselves.

1.2 Problems and aim

The problem raised by the drafting is; how do does a player determine what hero is the best to pick? The way the player makes this decision is by so called counter picking his enemy. An example would be the hero known as Bounty Hunter, he wins on average 50% of the games that he is picked, but when he is played against another hero called Naga Siren, he only wins 46% of the time despite Naga Siren on average winning less than 50% of all her matches. This means that Naga Siren is a counter to Bounty Hunter because when she is played on the opposite team Bounty Hunter has a lower chance of winning than what he usually has. [3] Most heroes have proximately 5% difference in win rate when played against their strongest counter however some heroes are very susceptible to being counter picked and have their win rate changed by more than 10%. In the given example it is quite trivial to compare the win probability of two heroes and determine which one is the better fit. However, the complexity increases significantly when including the other 4 enemy heroes of the game, additionally it is also important that the hero pick also works well together with the rest of the team. Given that there is a hero pool of 120 different heroes the number of possible team combinations is $\frac{(120C5 \cdot 115C5)}{2} = 1.4610^{16}$. This number is far too big to brute force every combination. However, this raises the question to what extent can deep learning be used to draft accurate heroes in Dota 2? To explore this question this project aims to use deep leaning to create a model that can be used to predict the best hero to pick given a draft from a Dota 2 game using information that a player can expected to know at the final pick of the draft.

2 Contributions

The demand for this type of product exists in both professional and amateur play. In Dota 2 tournaments a single coach is allowed to help the team during the draft. while the coaches are not allowed any digital help during the draft, this product can server as very good preparation. Additionally, amateur player who just want to increase their winning chances can use this product in order to gain an advantage. The betting scene for professional matches is also very big for Dota 2 and an individual could this product to evaluate the draft of either teams in order to place bets on which teams would win to increases the chance of winning the bet. It is also possible to branch out the concept to other games within the same genre for example League of Legends, Heroes of the Storm, Smite, and other MOBA games. It is to note that the game Dota 2 is dynamically updated every few months where heroes are changed, and new heroes are added. This means that, for the project to keep its relevancy new It needs to be constantly updated and retrained. This project does raise an ethical concern that it may be viewed as unfair to other players to use an AI to assist with the drafting. In order to ensure that this project does not the project in its trained from will not be publicly released.

3 Implementation

3.1 Requirements

Initially these are the requirements of the project:

1. The project needs to be able to fetch data samples from Dota 2 matches.
2. The fetched data needs to include matches from various skill groups
3. The project needs to be able to discard samples that are not useful (matched with irregularities).
4. The project needs to be able to reformat the samples and make it applicable for machine learning.
5. The project needs to use machine learning to evaluate the samples based on the draft.
6. The evaluation needs to have an accuracy that is equal or better than an experienced human.
7. The project needs to be able to input a sample manually to be able to evaluate matches that are hypothetical or yet to be played.

3.2 Design

The design of the project will be divided into smaller parts to ensure that the project does not become too complex. Each part deal with a specific set of requirements.

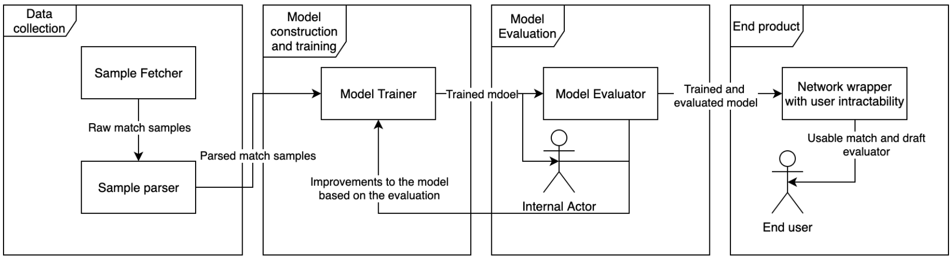

Figure 1: Project Design

Figure 1 illustrates the way that these requirements will be implemented. The project will be split into three main parts that all have different purposes in the project as a whole and the able to fulfil each requirement.

3.2.1 Data collection

Collected matches that will be used for the training of the network. The matches need to be played by both high- and normal skill players. They also need to be played by real players, not matches that are intentionally lost, contain bots or players who prematurely leave the game.

3.2.2 Model construction and training

The model construction will consist of creating the nodes of the network and to reformat the data in a way that the model can efficiently interoperate. This part will also consist of the training of the model using the samples from the data collection phase.

3.2.3 Model evaluation

This phase will consist of the evaluation of the accuracy of the model. It will examine how well the model can evaluate a sample of matches compared to experienced humans. It will also asses what parts of the data that the network values as reliable. The evaluation will then be used to make improvements to the way that the data was parsed by for instance removing faulty samples, add more samples to reduce selection bias, or to increase the sample size. There may also be some adjustments to necessary to make to the network more accurate, such as changing the number of nodes or the number of hidden layers.

3.3 Architecture

3.3.1 Data Collection

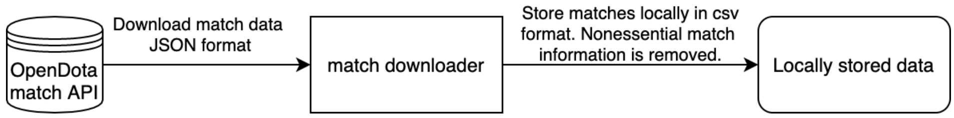

Figure 2: Data collection Architecture

The data collection is explained in figure 2. The project uses the OpenDota api (not to be confused with OpenAI) to download matches. This is a third-party service and not the official api provided by Valve Software. The reason this api was chosen is because OpenDota provides a greater access of past matches enabling the download of a greater number of matches. This has allowed for the download of just under 2 million matches for this project. That is approximately one week worth of matches. [4] OpenDota also allows for the download of match summaries, this means that just the essential information is contained such as the draft, match duration and rank of the players involved while leaving out information that is too match specific and not relevant to the project. This meant that the download of matches became a lot faster as OpenDota was tested to download approximately 100 matches per second compared to official Dota 2 Api provided by Valve Software which only allowed 10 matches per second. All of this led to the decision to use the OpenDota API for the data collection.

The locally stored data was formatted in a comma separate value format. The first value was the match-id which is a unique identifier for each match which can be used for debugging and in the evaluation of the network. The next 10 values are the heroes that played in the match sorted by team. Internally heroes also have a unique id that was used rather than storing the full name. The next value stored is the duration of the match in seconds, this value is useful since the duration of the match has a significant impact on what heroes that preform the best. The second the last value is the rank of the match, an integer between 15 and 80 with 80 being the more skilled players. The rank is important since some heroes require a lot more skill to play well and would thus perform better at higher ranks. An example of this is the hero Lycan who has a win probability of above 60% in the high skilled games while only having a win probability of 45% in low skilled games. [5] Lastly the label is stored to determine which team that won the game.

3.3.2 Model construction and training

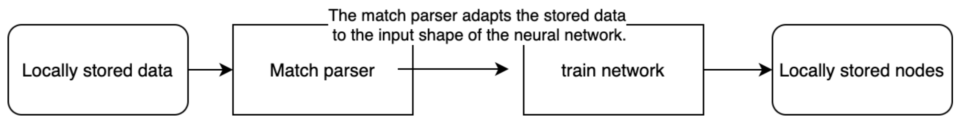

Figure 3: Data parsing Architecture

In figure 3 the stored data is reformed to the shape of the neural network. The input shape is 259 nodes. Two nodes for each hero to indicate its presence in either team, and two more need the match duration and rank and side of primary team. With only 120 heroes this adds up to 243 nodes, however there are some hero-ids that are not meaning that the highest hero id is 129. Thus, there are some nodes in the model that are not used but the convince of being able to index the nodes by hero-id is outweighs the inefficiency of a few more unused nodes. The network only has one output layer which is the evaluation of the match. This value tends towards 0 if the primary team is expected to lose and towards 1 if the primary teams is expected to win. This means that each downloaded match is used twice, once from the perspective of the winning team and once from the perspective of the losing team. A value close to 1 or 0 is a very confident evaluation but a value that is closer to 0.5 (the expected value) is a less confident evaluation. The model uses three hidden layers with 158 nodes each. The reason three hidden layers is simply because when evaluating it was the number of layers that yielded the best. Additional layers may be more accurate if a larger training sample could be provided. However, the returns gained by added more layers are expected to be demising while training time grows exponentially. Likewise, 158 nodes per hidden layer also gave the best results when tested. 158 being half the since of the input layer, which follows recommendation for neural networks using only once output node. [6]

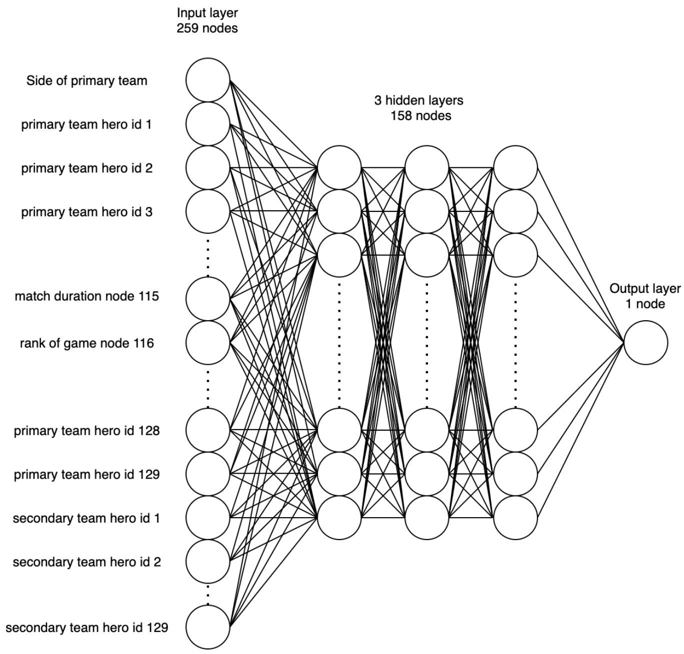

Figure 4: Neural Network Architecture

Figure 4 shows an illustration of what the final neural network looks like. In summary the input layers have two nodes for each hero id, one for each team. These nodes take the value 1 if the hero is being played on that team and the value 0 if the hero is not played. Some hero-ids are not in use for example hero-id 115 and 116. These are instead used for the match duration and the rank of the game. The three hidden layers have 158 nodes each making them half the size of the input layer. Lastly the output layer has a single node which takes the value 1 if the network predicts the primary team will win the match and take the value 0 if the network predicts the primary team will lose the match.

3.3.3 Model evaluation

In order to evaluate the model, the value of the output node needs to be discussed. As stated, a value close to 1 indicates that the model predicts a win and a value close to 0 predicts a loss. However, this raises the question how a value of 0.52 is meant to be interpreted. In that scenario the network does not predict strongly in either way. Hence for simplicity, the accuracy of the model is defined as the percentage of predictions that are correct when all predictions greater than or equal to 0.5 is a predicted win and all predictions less than 0.5 is a predicted loss.

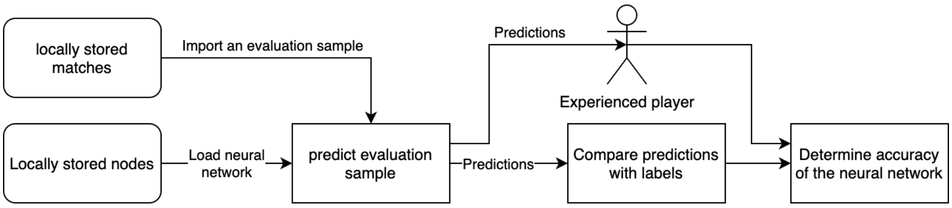

Figure 5: Model Evaluation Architecture

Figure 5 illustrates the way that the model was evaluated. It is to note that different samples were used in training and evaluation. The evaluation sample consisted of about 100000 samples randomly selected and excluded from the training data. This data amount should be sufficient to give a good estimate of the accuracy of the model.

3.3.4 Results and Evaluation

As stated earlier, the final product had an accuracy of 61%. However, this does not show the full picture. Since the prediction made by the model is variable, the matches that the model was more confident in its result tended to be more accurate.

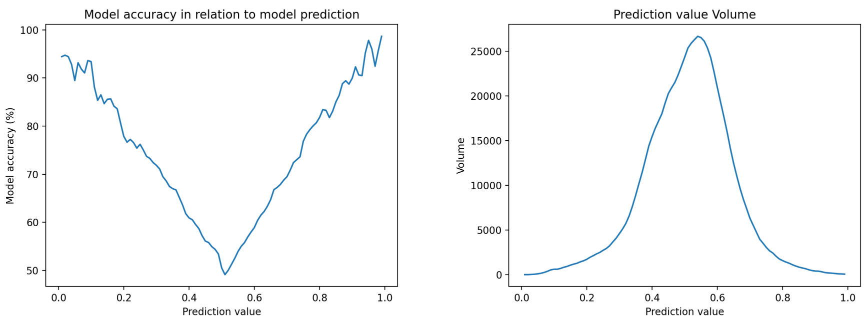

Figure 6: Predicted value compared to accuracy

Exactly this is illustrated in figure 6. In the left-hand graph, the x-axis is the prediction of the model rounded to the nearest hundredth and on the y-axis is the accuracy of the model for all evaluation samples that received that prediction. It is very evident that when the model gave a prediction close to 1 or 0 it was a lot more likely to be correct, likewise the closer the prediction is to 0.5 the less likely the model is to be accurate. This means that if the model makes a prediction of a value close to 0.5 such as 0.53, the value is does not carry much meaning as the likelihood of being correct is not much higher than chance. It is to note that there were a number of predictions that were greater than 1 or less than 0 but they were not included in the graph as there were insufficient samples in that range to make a reliable estimate of the model's accuracy. With this in mind, how many matches can then be predicted? That question is answered by the right-hand graph which plots the same value on the x-axis as in figure 6, however on the y-axis is the frequency of the prediction. It is every clear that the predictions follow a bell curve and that the vast majority of predictions are close to 0.5. In fact, 79% of all samples are predicted between 0.35 and 0.65. This means only a comparatively few samples could be confidently predicted in either direction.

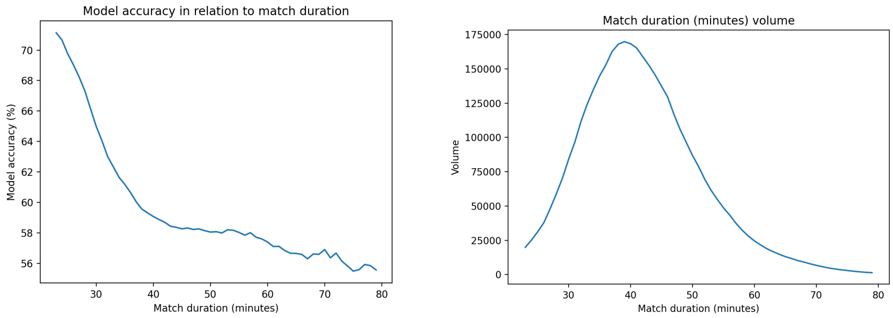

Figure 7: Match duration accuracy and volume

Additionally, one interesting aspect of the model is the correlation between the match duration and the model's accuracy. This is depicted by Figure 7 in which the left-hand side graph shows the duration of the sample matches on the x-axis and the model's accuracy on the y-axis. As seen the model is very accurate at predicting matches that end quick and becomes less accurate the longer the matches are. Given some thought this does seem very logical as a long match would indicate that neither team could quickly gain an advantage meaning that the draft was equal, resulting in that the winner of the match being be difficult to predict.

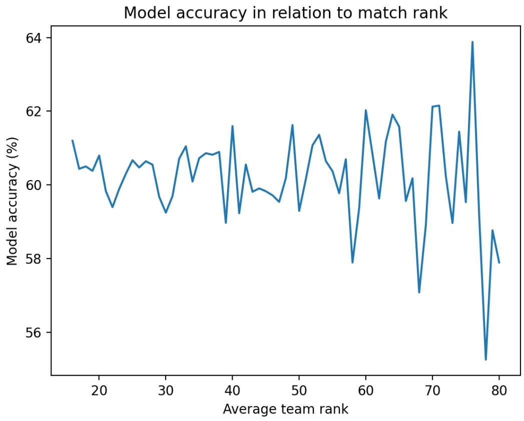

Figure 8: Rank and accuracy correlation

Figure 8 shows the relationship between the rank of the match and the model's accuracy. Surprisingly there does not seem to be any obvious relationship between the two. Perhaps this is because the training was done with samples of all different ranks and the network is slightly underfitted. In that case training with samples limited to a smaller rank range would make the network more applicable to a certain rank rather than showing no correlation at all.

4 Conclusion

To answer to what extent can deep learning be used to draft accurate heroes in Dota 2? In short about 61% of the time. This number does not seem very impressive being only 11% better than a coinflip. However, this also does go to show that Dota 2 as a game is still a skill-based game and that the draft does not single handedly determine who is the winner, but rather the gameplay after the draft. This is most likely intended by Valve Software as games where the draft plays a big role in who wins the game are usually deemed as unfair or unbalanced. In addition to this the use of AI in assistance of draft does not account for the individual skill of a player. For instance, if the AI suggests a certain draft would give a team a 5% advantage, it does not account for potential decrease in advantage due to the players playing a draft that they may be unfamiliar or less skilled with. With all this considered letting AI draft live games may not be worthwhile yet, however used as an assistance it can be useful to make suggestions on what to draft and what not to draft but leaving the ultimate choice to a human actor.

5 References

1. OpenAI. (2020, September 02). Dota 2. Retrieved January 10, 2021, from https://openai.com/blog/dota-2/

2. OpenAI. (2020, October 05). OpenAI Five. Retrieved January 10, 2021, from https://openai.com/blog/openai-five/

3. Bounty Hunter. (n.d.). Retrieved January 10, 2021, from https://www.dotabuff.com/heroes/bounty-hunter

4. Heroes - Most Played, This Week. (n.d.). Retrieved January 10, 2021, from https://www.dotabuff.com/heroes/played?date=week

5. Dota 2 Statistics. (n.d.). Retrieved January 10, 2021, from https://www.opendota.com/heroes/public

6. Keim, R. (2020, January 31). How Many Hidden Layers and Hidden Nodes Does a Neural Network Need? Retrieved January 10, 2021, from https://www.allaboutcircuits.com/technical-articles/how-many-hidden-layers-and-hidden-nodes-does-a-neural-network-need/