# AI Applied to Games Report

Sebastián Molina López, Unai Zabala Cristobal

October 2018

# Contents

# 1 Introduction

When nowadays we speak about AI in games, lots of us think on video games but we can also speak about board games like chess or Go. The most popular agents in history are Deep Blue, who won against the world best chess player and AlphaGo, the neural network that beats Lee Sedol, world best Go player.

## 1.1 Deep Blue

This IBM's machine of 1996 is one of the first machines trying to beat humans in games as we know, and it combines four different AI techniques. First of all in creates a model using a tree search, where each state is a particular arrangement of the pieces on the board and the available actions correspond to the legal chess moves for the current player in that arrangement.

Once the model is done it is the time to apply algorithms. First of them it is not an algorithm by itself, is the evaluation function. The main goal of the function is going to be telling us which state to go for in the tree, using a mathematical function that should evaluate how good is each movement we can do in the state we are. We would probably want our function to give high values when opponents king is in checkmate.

Given the evaluation function value is time to apply some algorithm, and the minimax algorithm was the chosen one. It uses minimax algorithm, who serves for looking our best value given by the evaluation function and also takes into account the opponents movement.

At that moment the system should work and is time to apply some heuristics and optimizations. The main objective is to reduce the computation time and make it much faster and effective, for example using alpha-beta pruning, an algorithm that cuts some branches of the tree.

## 1.2 AlphaGo

Here, instead of explaining how it works like in the Deep Blue case, we are going to tell the main differences or innovations compared with the previous one. The main idea was quite similar, creating a tree search and applying algorithms.

Instead of a minimax algorithm with alpha-beta pruning it uses Monte Carlo Tree Search (for now in on MCTS), a more refined algorithm. In MCTS the tree expands on the most promising moves until it reaches a final move (win/loss/draw), simulates them and get a value or weight (like before, using a evaluation function).

# 2 Neural Network Applied to Dota2

## 2.1 OpenAI Five

OpenAI Five is an AI compose by five neural networks. It is now defeating skilled amateur teams at Dota2. Even if at the beginning it plays with restrictions, the aim is to play against pro teams and with no restrictions (managing a limited set of main characters) in The International.

The machine needed for 5v5 is stronger than the 1v1 version. OpenAI Five plays against itself what would take 180 years of games to human in a day, learning by self-play. It trains using a scaled-up version of Proximal Policy Optimization running on 256 GPUs and 128,000 CPU cores. Each one of the five heros, is represented by a LSTM network (a Recurrent Neural Network composed by Long Short Term Memory units) and with no human data, it learns recognizable strategies.

Now we are going to mention some of the fields where OpenAI Five must master and where it haves some problems.

- **Long time horizonts.** Dota run at 30 frames per second for an average of 45 minutes. Most actions (like ordering a hero to move to a location) have minor impact individually, but some individual actions like town portal usage can affect the game strategically; some strategies can play out over an entire game. OpenAI Five observe every fourth frame, yielding 20 000 moves. Instead of 40 moves that Chess match usually ends, or 150 moves by GO match, with almos every move being strategic.

- **Partiallly-observed state.** Units and buildings can only see the area around them. The rest of the map is covered in a fog hiding and their strategies. Therefore a strong play requires making inferences bases on incomplete data.

- **High-dimensional, continuous action space** In Dota, each hero can take dozens of actions, and many actions target either another unit o a position on the ground. For instances on chess there are a average of 35 actions possibilities, in go 250, but here there are a average 1000 around.

- **High-dimensional, continuous observation space** Dota is played on a large continuous map containing ten heroes, dozens of buildings, dozens of NPC units, and a long tail of game features such as runes, trees, and wards. Our model observes the state of a Dota game via Valve's Bot API as 20 000 (mostly floating-point) numbers representing all information a human is allowed to access. A chess board is naturally represented as about 70 enumeration values (a 8*8 board of 6 pieces types and minor historical info); a Go board as about 400 enumeration values (a 19*19 board of 2 pieces types plus Ko

The system entirely learns by itself, starting with random parameters and using reinforcement learning. In *Table 1* you can see the numbers of the system.As we do in the introduction, let's try to do a general view of how it works.

For this case we have not any exploration tree, no minimax either MCTS, it would be near impossible or impossible to handle it because of the amount of scenarios we can find in the while of the game. Instead of that, we have five single-layer, 1024-unit LSTM that sees the current game state and emits actions

| | OPENAI 1V1 BOT | OPENAI FIVE |
|---|---|---|
| CPUs | 60,000 CPU cores on Azure | 128,000 pre-emptible CPU cores on GCP |
| GPUs | 256 K80 GPUs on Azure | 256 P100 GPUs on GCP |
| Experience collected | ˜300 years per day | ˜180 years per day (˜900 years per day counting each hero separately) |
| Size of observation | ˜3.3 kB | ˜36.8 kB |
| Observations per second of gameplay | 10 | 7.5 |
| Batch size | 8,388,608 observations | 1,048,576 observations |
| Batches per minute | ˜20 | ˜60 |

Table 1: Numbers and specs of each OpenAI machine

through several possible action heads. The main reason for using LSTM networks is that are explicitly designed to avoid the long-term dependency problem, being able to learn long-term dependencies.

Using neural networks means that we have a graph. The evaluation function (that is going to be much more complex because of the amount of little things to take into account) is going to be much more complex, and its objective is going to be to change the weights of the graph constantly.

In this case we have not a tree, so the exploration is no going to happen there. The exploration tooks place in the map, where it is going to analyze hundreds of items, dozens of buildings, spells, and unit types, and a long tail of game mechanics to learn about.

### 2.2 How does OpenAI Five work?

The AI works by reinforcement learning, so by the time it plays games it learns the more. It starts from far below human level and given good information and enough computation it reaches superhuman level.

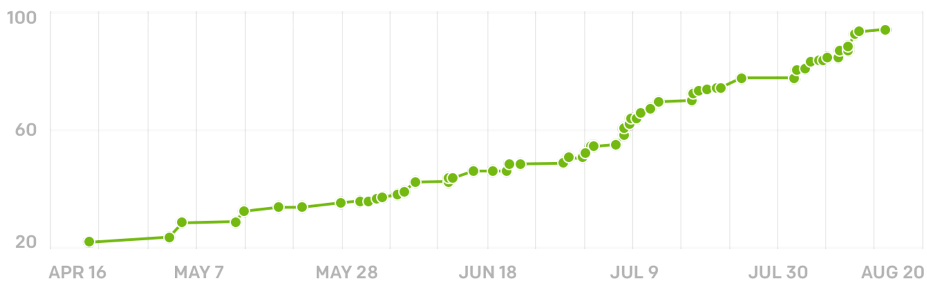


Figure 1: The image above shows us the TrueSkill rating through the time. TrueSkill is a skill-based ranking system developed by Microsoft for use with video game matchmaking on Xbox Live for more than two players. That values where gotten by simulating games between the bots and observing the win ratios and adding new features to algorithmic improvements, scaling things up...

As we mentioned before, the AI is able to play in 1v1 and 5v5 modality. It does not need any change for each type of game, it works automatically without any modification. To do that, lets see the following interfaces the bot operated in:

- **Observations.** The bot can observe the same features than a human can do. That way it is supposed to be in equal conditions.
- **Operations.** Trying not to be in advantage against humans, the frequency of the operations is limited to the one that humans have.
- **Feedback.**The bot received incentives for winning and basic metrics like health and last hits.

### 2.3 OpenAI Five vs Dota2 Pro Team

Actually, they are applying AI in a different way than mention before. In August of 2017 humans grab victory in Dota2 against OpenAI Five, an AI compose by five neural networks. In the first half of the match, there were continuous changes on taking the lead but at the end human strategy was the winning one.

OpenAI Five plays quite well, but its main mistake was to paying too much attention to Roshan, a risky strategy. He is the one of the main characters of the game, he is ubicated in the middle of the map, and killing him gives serious rewards such as an item that allows heroes to respawn quickly. Sometimes works for the AI giving extra lives for their main characters, but other times they paid for it allowing the humans to dig in their undefended territory.

Apart from that errors the bots make a good game. As AI games researcher Mike Cook commented on Twitter: “The bots are still very good at moment-to-moment, but they seem bad on macro-level decisions.”

Even if OpenAI Five was defeated, they have previously won matches against skilled amateur players (5v5) and against professional players in 1v1 modality. That precedents were not significant because of their origin. The first of them was gotten against amateur skilled players and with lots of restrictions, and the second one in a 1v1 modality, where skills like precision timing are very important and where machines clearly wins.

For winning against skilled amateur players they use reinforcement learning and victories are always preceded by defeats. So the question now is, will the OpenAI Five learn from this game and win in a early future?

### 2.4 The International

In The International, the concluding tournament of the Dota Pro Circuit, they put Sumail (the bot) playing in LAN against humans. It was hard for them to beat Sumail but at the end they got it and they do it attacking the exploits you see below:

- **Creep pulling.** it’s possible to repeatedly attract the lane creeps into chasing you right when they spawn (between the bot’s tier 2 and tier 3 towers). You end up with dozens of creeps chasing you around the map, and eventually the bot’s tower dies via attrition.
- **Orb of venom + wind lace.** this gives you a big movement speed advantage over the bot at level 1 and allows for a quick first blood. You need to exploit this head start to kill the bot one more time.
- **Level 1 raze.** this requires a lot of skill, but several 6-7k MMR players were able to kill the bot at level 1 by successfully hitting 3-5 razes in a short span of time.

## 3 Summary

As a summary we can say that the evolution between different games is big. There is a double evolution. In one hand we have the computational advance due to better algorithms and new techniques. In the other hand we have the hardware, that is much more powerful and dedicated for AI tasks. We also have seen how difficult is to design and implement this kind of applications, with so many features to analyze and with need of very powerful machines.

AI seems to be near to achieve that superhuman performance, but we have to say that it occurs at very specific tasks. In the future we will like to see a machine or program capable to have superhuman performance y different areas and seeing the uses we can give to it.

## 4 References


- OpenAI reinforcement videos: https://sites.google.com/view/multi-agent-competition
- Multiagent competition code: https://github.com/openai/multiagent-competition
- Dota2 new: https://www-theverge-com.cdn.ampproject.org/v/s/www.theverge.com/platform/amp/2018/8/23/17772376/openai-dota-2-pain-game-human-victory-ai?amp_js_v=0.1#amp_tf=From%20%251%24s&ampshare=https%3A%2F%2Fwww.theverge.com%2F2018%2F8%2F23%2F17772376%2Fopenai-dota-2-pain-game-human-victory-ai
- OpenAI - More on Dota 2: https://blog.openai.com/more-on-dota-2/
- OpenAI Five: https://blog.openai.com/openai-five/
- Proximal Policy Optimization: https://blog.openai.com/openai-baselines-ppo/
- Understanding LSTM Network: http://colah.github.io/posts/2015-08-Understanding-LSTMs/#lstm-networks
- AlphaGo: https://deepmind.com/research/alphago/
- Reinforcement Learning: https://aithority.com/gaming/reinforcement-learning-really-works-for-ai-against-pro-gamers-openai-trailblazer-says/