AI coach for video games

Tom Rochette <tom.rochette@coreteks.org>

January 21, 2020 — b9925e2d

1 Question

How would you build an AI that could offer coaching for games like StarCraft or Dota/LoL?

2 Answer

I see coaching as similar to the loss function of a machine learning model. I also see coaching as trying to optimize a (program's) function by figuring out where the largest improvements can be made. In order to provide effective and useful feedback, a coach should focus on the areas where the player shows the most potential for improvement. In a game like Starcraft, that would mean first pointing out the macro level mistakes then the micro level mistakes.

To coach you need to have a model of which actions have an impact on the game and how much impact they have. A learning algorithm/model like AlphaZero generally exhibits two types of learning: learning from observation and learning by playing against itself.

2.1 Learning from observation

2.1.1 Humans

The most common approach to coaching is by observing more successful players than themselves. The learners may watch better players while the better players are playing the game and commenting on their gameplay or the learners may watch someone reviewing a replay and providing their own analysis. Both of these cases can be seen as models (the players) trying to explain their internals (the logic behind their actions).

Coaching generally starts by trying to reproduce the recipe of someone else. You may not understand why they are doing certain things, but you do it yourself and you observe the results. As you practice repeating those same observations => actions, you try to reproduce as closely as possible what the better player would do.

2.1.2 AI

In the first learning phase, the model simply observes what happens during gameplay. In competitive games such as MOBA/RTS, the only reward signal is the victory/loss at the end of a game. As human beings, we quickly learn that winning a fight/encounter is good and losing it is bad. Those give use intermediate reward signals that an AI agent may not be able to build right away since it is conceptually difficult to determine when an encounter begins and ends. The agent could however learn a simple metric such as *the sum of the health of all units*, where keeping this value high is generally a good thing.

The model will need at some point to be able to establish its own scoring system so it can give itself some intermediate rewards during a game. It will also need to learn how to segment a sequence of actions into repeatable action units such as constructing unit X, attacking player Y, defending zone Z. As such, it may deem that constructing unit X is worth 5 units of reward, attacking player Y is worth no reward and that defending zone Z is worth 30 points of reward. The value of rewards may vary based on numerous factors,

such as how much time has elapsed since the beginning of the game, the known enemy army composition, existing vision, etc.

Having actions such as "attack coordinate X, Y" are a too low level. Your model will have to learn hierarchically complex actions such as "attack player X", "attack the gatherers of player X", "attack the weak gatherers of player X", etc. which will then translate down the hierarchy to "attack unit at coordinate X, Y".

An AI coach may look at hundreds or thousands of replays and observe the distribution of units allocation after 1, 2, 3, 5, 10, 15, 20 minutes (or every 5 seconds) into the game and their correlation to whether the player won or lost. It may look at the items purchased by the player in a MOBA game, their timing and their correlation to whether the player won or lost. For a human being to do similar thing would require a lot of time. Most would probably write scripts to automate the process of collecting those details instead of manually going through the replays one by one.

2.2 Learning by playing against yourself

Playing against yourself is more complicated. A perfect recording of your actions may not prove difficult to beat. It may send units to the wrong location on the map, be caught off guard moving to a location while you positioned units in the middle of the path, it may react to an attack the "replay" opponent had sent to its base at one point in the game, etc. It is however a start, one example you can train against.

A lot of players who are invested in the game will do theorycrafting which is basically to use logic and reasoning in order to assess what to do in specific situations. They simulate various potential cases in their head and they devise plans to defeat them. While a human being may be able to devise a few dozen simulations over an hour, a computer may be able to generate hundreds of thousands. It may also be able to test them in a more accurate simulation environment. When game patches are released, it could rerun all the simulations it had generated to determine the impact of the patch on its existing strategies.

An AI coach can be provided the game rules, specifically, which units are weak/strong against other units, and look at the game while you are playing. If you attack your opponent and the AI observes a strong concentration of a specific type of unit, and it notices you do not have any of the units that counter this unit type, it may suggest that you start building those as soon as possible. It may also notice that your unit composition is weak against the unit composition of your enemy and suggest units to build to balance your army and to be better prepare for the next encounter.

We can see this act of theorycrafting as the equivalent of knowing, at a high level, the strategies and counter-strategies one can employ at an early point in the game, the same way you can learn the different opening moves in chess.

In the case of learning by playing against yourself, what we want the AI coach to provide us is an opponent that will challenge our current biggest weaknesses so we can address them. In many cases certain specialized strategies will be extremely strong against a specific type of strategy and we will want to know those cases so we can use those strategies when the time is right.

2.3 Features we may want from an AI coach

- Determine your weaknesses/areas of improvement
- Suggest potential approaches to solve recurrent problems we have
- Suggest heuristics that may be easy to understand and follow as human beings
- Simulate opponents that would exert your current weaknesses so you can practice against them
- Collect various gameplay related statistics their associated success rate (number of units of type X after Y minutes, number of creeps killed after X minutes, items purchase order, build order, etc.)

3 Things to explore

- If you were in an environment where you had access to very few replays, how would you learn the most out of those available?