Sean Steinle, Kenan Rustamov, Gordon Lu, Jake Baumbaugh

CS1678

February 4th, 2021

Winning Pokémon Battles with Reinforcement Learning

**1.What do you propose to do?**

Pokémon is a video game that has been extremely popular for several decades. The game revolves around training Pokémon--animal-like creatures--and battling other trainers and their Pokémon. In this project, we’d like to focus on just one aspect of the game: winning Pokémon battles. Within this goal, there are levels of complexity we can introduce. After succeeding at each sub-task, we can move onto another. For example, we will start with just two Pokémon, with just a few moves each. Once our model can successfully carry out a battle, we can add more factors like different sets of moves, different Pokémon, or even disparity in Pokémon level (which roughly corresponds to difficulty level).

**2.** **What have others attempted in this space, i.e. what is the relevant literature?**

A paper from undergraduates at Stanford reveals that there have been a few attempts at using machine learning for winning Pokémon battles. These papers used reinforcement learning to achieve a win rate of 65% and the highest elo achieved was around 1300. This means that the algorithm was only slightly better than the average player. These algorithms can be optimized to yield higher win rates through algorithms that track more variables.

**3.Why is what you are proposing interesting?**

This task is interesting because it has natural ancestors in elder artificial intelligence models such as Deep Blue and Alpha Go. Similar to engineering a model to play chess or go, this task involves the model learning the structure and strategies of a game with a relatively limited domain. This task is also interesting for our team and generation especially because this game was exceedingly popular with children and teenagers in the late 1990’s and early 2000’s. There should be plenty of challenge for the algorithm due to the numerous factors that go into a battle including stats, types, items, moves, and levels.

**4.Why is it challenging?**

This task is challenging because reinforcement learning is challenging. We do not have a dataset of successful Pokémon battles. Instead, we have basic information about Pokémon and the ability to simulate battles. Moreover, the task is challenging because of the lack of research into this specific task. There is also an issue that lies in retrieving relevant data for the Pokémon battles. There are many different data points that we can collect on the current team and the enemies team. These include stats on the Pokémon, the Pokémon itself, the movesets, the levels, and the types of the Pokémon. These are not all readily available in a Pokémon battle, so a work around will have to be formed. As far as we can tell, there have only been two other teams to journey to this horizon of modern science.

**5. Why is it important?**

After being defeated by Deep Blue, several grandmasters described the opposing player as “a wall coming at you”. Some may not consider this task to be important. After all, Pokémon is just a game, and it is a game without the several millennia history of chess or the notoriety of go. However, creating artificial intelligence games plays a special role in advancing and regulating artificial intelligence. Creating models that engage in such a jovial part of our lives can lead us to understand the divide that exists between humanity and such an increasingly important technology.

**6.What data do you plan to use?**

In regards to data use, we are in a unique position: because we are using reinforcement learning, we will not be utilizing the more traditional supervised learning approach of having and predicting labels. Instead, we’ll be using the output of our neural network as its own input. We’ll supplement our model with two datasets about the Pokémon universe, so that our model has access to important features like Pokémon type, Pokémon moves, and other attributes.

**7.What is your high-level idea of how your method will work?**

As mentioned earlier, we’ll be using reinforcement learning. This entails using our network’s output as its own input. As our model trains with randomly-generated Pokémon teams, it will learn from its prior games to improve going forward. To achieve this goal, we are going to need to create a Pokémon battle simulation that can take neural network output as its player inputs, as well as create a trainer AI to battle against. We can explore many options for this, including a random AI, trying to replicate the AI from the games, or even a minimax AI, if we feel that our model needs a challenge.

**8.In what ways is this method novel?**

This task is novel because previous popular approaches to similar tasks often did not use deep learning. For example, Deep Blue, the chess computer from IBM that beat grandmaster Gary Kasparov, primarily used databases and brute search methods to beat the world champion. Likewise, Alpha Go, the model from DeepMind that has taken on several grandmasters in the game of go, also uses a Monte Carlo search algorithm--though it does incorporate some deep learning. Thus, our model will be another step forward in solving a limited domain game with deep learning.

**9.How will you evaluate the method, i.e. what metrics are you going to use, and what baselines are you going to compare to?**

This task is somewhat difficult to measure because of the aforementioned lack of exploration. However, we know that one team out of Stanford achieved a 65% success rate under random pokemon battle conditions. We will also be using the percentage of games won in a random battle as the evaluation for the algorithm. We can also take it one step further and have the AI battle players online which would give us more accurate data on success rate and give us an ELO to determine efficiency of the algorithm. A good baseline is likely greater than 50%, as the computer-generated trainers in the game are fairly simple and easy to defeat. An ELO to attempt to beat would be around 1000.

**10.Give a (1) conservative and (2) an ambitious schedule of milestones for your project.**

We propose to have two, week-long reading periods, followed by two-week long sprints for development. Here is our schedule, if things go according to plan:

* 2/4 - 2/16: read about reinforcement learning, implementations, other pertinent design details
* 2/16 - 2/23: decide on structure and evaluation of network
* 2/23 - 3/9: engineering, have network playing Pokémon by 3/9
* 3/9 - 3/23: engineering, have network winning battles; look into new challenges
* 3/23 - 4/6: engineering, have network attempting new challenges
* 4/6 - 4/22: analysis, examine and interpret network success on various subtasks

Here is our schedule, if things do not go according to plan:

* 2/4 - 2/16: read about reinforcement learning, implementations, other pertinent design details
* 2/16 - 2/23: read more, decide on structure and evaluation of network
* 2/23 - 3/9: finish deciding, start implementation
* 3/9 - 3/23: engineering, have network playing Pokémon
* 3/23 - 4/6: engineering, have network winning Pokémon battles
* 4/6 - 4/22: analysis, examine and interpret network success on task

Important dates:

* 3-2: Report I due (lit review, detailed plan for implementation and evaluation)
* 3-23: Report II due (development check)
* 4-22: Report III (finish)