

Reinforcement Learning on Pokemon Emerald’s Battle Tower

Rishabh Jain, Samarth Venkatesh
University of Washington

1. Introduction

This paper examines implementing reinforcement learning to single-player challenges in Pokemon Emerald. Reinforcement learning has previously been successful at learning how to accomplish tasks such as walking [4] and even playing games [5].

Previous attempts to use machine learning (ML) techniques on Pokemon have focused primarily on either on the PvP aspect [3] or playing through a game in its entirety [6]. We attempt to explore a specific part of single-player Pokemon that fuses these two aspects together; the Battle Tower. In the Battle Tower, the player attempts to keep winning consecutive battles with their “streak” ending with a loss. Since the only actions available in this challenge have to do with battling, we could potentially gain insight into how a given RL algorithm learns and optimizes a subset of Pokemon. Any revelations could be adapted and applied to Pokemon in a larger context or even other similar games.

Due to the smaller scope of this application as compared to player-vs-player or playing through the whole game, we hope to achieve consistent results in winning a few battles without large training time, which may reflect a potential in using reinforcement learning to maximize a streak. In particular, we hope to be able to get a streak (defined as 7 consecutive victories in the Battle Tower with the same team). We later hope to train our model to win the Silver Symbol (5 consecutive streaks) and Gold Symbol (10 consecutive streaks) as these are tough challenges for a regular trainer.

2. Related Work

Reinforcement learning (RL) is a popular tool in training ML models to succeed in environments where information is incompletely known. The first major instance that demonstrated RL’s viability for training agents to exceed human experts at a game with fixed rules was Mnih *et al.*’s paper “Playing atari with deep reinforcement learning” [5].

The paper demonstrates that the principle challenges with RL algorithms is the timestep disparity between a given action and its resultant reward and the “sequences of highly correlated states” that the algorithm encounters

rather than the independent data samples inherent to other models. The solution to this, the paper found, was to use a convolutional neural network to learn successful control policies and then smooth out the training distribution by sampling previous transitions.

Within Pokemon specifically, Sihao Hu *et al.* [3] show demonstrate the viability of such an approach, using LLMs to autonomously play Pokemon and use in-context RL to refine policy when making decisions. Pablo Barros *et al.* [2] use a 3 layer neural network model to learn to represent a battle and is trained using composed loss based on contrastive optimization to train a model to play effectively against specific opponents.

Finally, there are a few hobbyist explorations of designing ML models to play Pokemon battles [6] and complete the single player game [1], which use various ML models including RL but due to the dynamic nature of a Pokemon battle and the many orders of magnitudes of possible game states, it is difficult to train an effective model. In addition, these models are effectively trained from scratch and as a result “waste” resources in re-learning policies that may have been investigated before. In addition, due to these models not being trained directly on the environment itself, they may not find the optimal policy for battling within the confines of the Battle Tower.

3. Technical Challenges

A primary hurdle for us to overcome before training can commence is the connection of our model to the game itself. This will involve using OpenAi’s Gym Retro and a GameBoy Advance Emulator to interact with the game using Python. From there, we can begin creating and using classes for the game to represent states, agents, etc. and begin training the model.

Another challenge is the volume of data required to train a model to effectively play a Pokemon battle along with the number of parameters to tune. Due to several factors such as team composition (of both the player and the opponent), random damage multipliers, number of possible moves a Pokemon can execute among many other details, it will be difficult to tune parameters effectively and decorrelate the game states and resulting victory probabilities. We ex-

pect that model free algorithms such as PPO or Q-Learning would be more effective here due to requiring less computation and less data, while model based algorithms such as minimax would result in easier tuning.

4. Experiments

Our model will center around maximizing a streak with a set team of Pokemon. This removes the added complexity of team-building, training, and catching for the model.

In addition, we plan on starting off a model free algorithm so as to reduce the computational overhead associated with training a model on a highly dynamic environment and use data efficiently. We later hope to compare the effectiveness of model free and model based algorithms to determine which set of algorithms works best in the context of a highly dynamic environment with a fixed win condition. To compare their effectiveness, we can simply compare the maximum, top 5, mean and lowest streaks achieved by each model to determine not only which model achieves the greatest streak, but also which model is consistently the best at achieving a streak.

To visualize these, it may be helpful to plot the average return per epoch against the number of epochs as well as the embeddings our RL model produces (which may also help in identifying the hyperparameters of our model). We also may plot the frequency of our model taking some action (using a certain move, switching, etc.) in a given turn and the number of times a given Pokemon in the party “faints” per battle to evaluate the effectiveness of our model.

References

- [1] Caleb Compton. How i taught an ai to play pokemon emerald. 2020. [1](#)
- [2] Pablo Barros et al. All by myself: Learning individualized competitive behaviour with a contrastive reinforcement learning optimization, 2023. [1](#)
- [3] Sihao Hu et al. Pokellmon: A human-parity agent for pokemon battles with large language models, 2024. [1](#)
- [4] Tuomas Haarnoja et al. Learning to walk via deep reinforcement learning., 2018. [1](#)
- [5] Volodymyr Mnih et al. Playing atari with deep reinforcement learning, 2013. [1](#)
- [6] Peter Whidden. Training ai to play pokemon with reinforcement learning, 2021. [1](#)