Comparing Application of Reinforcement Learning Algorithms to Pokémon Battles

Group 17 Member Info

Vadim Boshnyak (101151803) vadimboshnyak@cmail.carleton.ca
Peter Bailey (101157705) peterkbailey@cmail.carleton.ca
Safwan Wadud (101148932) safwanwadud@cmail.carleton.ca
Hamza Osman (101148649) hamzaosman@cmail.carleton.ca

Problem Statement

A Pokemon battle consists of two Pokemon Trainers pitting their teams of up to 6 Pokemon against each other in strategic, one-on-one, turned-based combat. Typically battles consist of the player battling against a non-playable character (NPC). Each individual Pokemon has features such as their type, health points (HP), statistics like their defense or attack strength, and a move set of 4 attacks with their own respective effects like lowering HP or other stats. An attack may be more or less effective depending on the type of move, the type of the opponent, and the attacker and defender's various statistics. A single Pokemon is in combat at any given time, trainers can sub them in and out so long as their HP is above 0. A trainer is defeated in battle when all of their Pokemon have been knocked out.

In our project the agent will act like the player, making use of a team of Pokémon to overcome their opponent in battle. Initially we will begin with the player and the adversary having a fixed team of 3 Pokémon each and the agent will learn to defeat the (static, unlearning) AI opponent. As we proceed with the project we hope to increase the complexity by having the agent learn to compete in a variety of battles with the ability to generalize similar states.

Feasibility

A Pokémon battle fits into the MDP framework as it can be broken into discrete actions, states, and rewards. Reinforcement learning algorithms can then be used to create an agent seeking to maximize its reward. A pokemon battle consists of an ordered sequence of moves and so finding the optimal sequence translates into finding the optimal policy.

States

The size of the state space for a single Pokémon battle is enormous, but it can be represented compactly with a discrete number of features.

- There are between two and 12 Pokémon, each with:
  - 1 or 2 of 15 elemental types
  - The following integer value stats: Health Points, Attack, Defense, Special Attack, Special Defense, Speed, and Accuracy
  - One of 9 status effects, including healthy
  - 4 different moves, each with:
    - 1 of 15 elemental types
    - The following integer stats: Power Points, Power, Accuracy
- Which 2 Pokémon are currently swapped in
- As we move forward with the project we may add complexity with additional features such as

    - The Field condition (i.e weather)
    - Each Pokémon could have one of 188 abilities
    - Each Pokémon could have one held item from a pool of the most

Actions:
The agent can perform a maximum of 9 different actions depending on the state. The agent can choose to use any of the four moves of the current pokemon in battle. The agent can also choose to swap in one of their other Pokémon, that have not fainted, from their party.
Depending on the current state not all actions will be available, for example:

- The agent can not switch to Pokémon which have fainted
- If their current Pokémon has fainted they can only swap to a new Pokémon, and not use one of the current Pokémon's four moves
- If the agent has a team smaller than 6 the only swap actions available will be for Pokémon in their party

Rewards:

- +1 at the end of an episode if the agent won the battle,
- -1 at the end of an episode if the agent lost the battle
- We may also consider the following:
    - Intermittent rewards for defeating opposing Pokémon
    - A final reward proportional to the agent's overall performance in the battle, with possible factors including: number of enemy/friendly Pokémon remaining, health of remaining enemy/friendly Pokémon.

Milestones

Dates	Milestones
October 1 - October 14	- Find and implement pokemon game library
October 15 - October 28	- Create custom Gym environment including state, action, reward - Implement wrapper for game library so agent can interact with environment, even if only randomly - Environment demo on Oct 30th or Nov 1st
October 29 - November 11	- Decide on specific RL algorithms to use - Build functional agent with at least 1 RL algorithm
November 12 - November 25	- Complete agent with remaining RL algorithms - All members understand implemented code, the algorithms used, and their respective drawbacks
November 26 - December 10	- Prepare result demo demonstrating performance of the implemented algorithms - Write report and analysis - Result Demo on Dec 6th or 8th - Report due Dec 10th

References

[0] K. Khosla, L. Lin, and C. Qi, “Artificial Intelligence for pokemon showdown,” https://docplayer.net/63514819-Artificial-intelligence-for-pokemon-showdown.html (accessed Oct. 4, 2023).

[1] A. Kalose, K. Kaya, and A. Kim, "Optimal Battle Strategy in Pokemon using Reinforcement Learning," https://web.stanford.edu/class/aa228/reports/2018/final151.pdf (accessed: Oct. 4, 2023)

[2] L. NORSTRÖM, “Comparison of Artificial Intelligence Algorithms for Pokémon Battles,” https://odr.chalmers.se/server/api/core/bitstreams/b5fac289-1328-41b8-ad78-f89eb39fce30/content (accessed Oct. 3, 2023).

[3] R. Rill-Garc´ıa, “Reinforcement Learning for a Turn-Based Small Scale Attrition Game,” https://ccc.inaoep.mx/~esucar/Clases-mgp/Proyectos/2018/reinforcement-learning-turn%20%281%29.pdf (accessed Oct. 3, 2023).