

CS-5100: Foundations of Artificial Intelligence (Spring 2022)

Project Proposal – Pokémon Battle AI

Nisarg Patel

Problem Description:

Pokémon is a famous video game franchise where player owns creatures known as Pokémon and completes a storyline. One of the most important aspects of these games is Pokémon Battle, a 2 player turn based battle, which is also the format used to play the Pokémon Competitive. In a Pokémon Battle, each player has at most 6 Pokémon and each Pokémon knows 4 moves. Both players have 1 pokémon in front. At every turn, a player can either select one of the moves to attack the or switch to any of the other not fainted Pokémon. A player loses if every Pokémon of the player is fainted.

Each Pokémon has a variety of features like types, base stats(health (HP), attack, defense, special attack, special defense, speed), abilities, and held items. Each move also has features like type, damage, accuracy, power points, etc. Currently there are more than 800 Pokémon and 800 moves in the Pokémon games. Types of moves and Pokémon determines whether a move with a particular type will do resisted, normal or super-effective damage to the Pokémon's type combination.

Given the player's Pokémon and moves that each Pokémon knows, this project aims at finding the optimal strategy to maximize the probability of winning. Considering that actual Pokémon Competitive has a lot of factors and thus state space increasing exponentially, I will be including many constraints like only damaging moves, no abilities, no held items and more. Also, there can be many variations of the problems, like opponent information hidden until used or no switching of pokémon, I will try to solve easier conditions first and expand the project to make the mechanics as close to Pokémon Competitive as possible. Computationally speaking,

Input: n pokémon ($1 \leq n \leq 6$) and their moves of player and opponent. With first pokémon facing each other.

Action: At each turn, player can attack using any of the 4 moves or switch to any of the (Pokémon whose Health > 0).

Terminal Condition: Game ends where every Pokémon of a player has 0 Health and the other player wins.

State: Each state would consist of current Pokémons of both player, moves known by each Pokémon, Stats(like Attack and HP) and types of current Pokémon of both player.

This problem is popular due to the vast options of combinations of moves, pokémon and their factors exist and Pokémon Competitive has also gained much popularity since its start. Along with it, there is only limited research done to create an optimal agent for Pokémon. The possible state space of Pokémon is extremely large just for a single turn considering all the pokémon and moves. Personally, I have played these games a lot and it would be really interesting to try to create an AI agent of the game myself.

Approaches/Algorithms:

For the problem, I will try to use the following approaches:

- 1) Epsilon greedy to select the move that does maximum damage with an epsilon chance of randomness.
- 2) Minimax which takes action based on the intuition that opponent will play its optimal action
- 3) Expectimax to factor in the chance by which the opponent will select the action.
- 4) Reinforcement learning(Q-learning).

The Evaluation functions that could be used in Minimax and Expectimax can be made by combining various factors of pokémon (like summing the health) in the state. Some of the studies on this problem have typically used these 4 techniques with best results coming as the AI playing as good as an average player based on the ratings maintained in Pokémon Showdown.

Planned Comparisons:

One of the comparisons will be the percentage of games won given the same initial state to all of the 4 agents with an opponent that select an action randomly.

Other comparisons could be made by change the evaluation functions within Minimax or Expectimax. The comparisons can also be made by changing the variables in Q-learning.