0 Contents

1 Results and Discussion

1.1 Training

1.1.1 MCTS

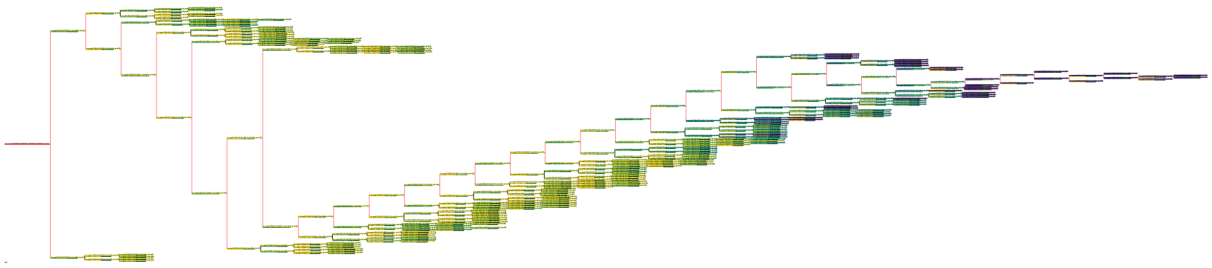

Figure 1.1: An example MCTS tree structure lasting 33 steps with 15 MCTS simulations per step. The neural network has not been trained and is working against an untrained adversary with and average power of 5% of the player. Lighter (yellow) colours represent "better" states.

Figure 1.1 shows the structure of a typical tree search when relying solely on the tree search. This is significantly better than with no tree-search, which typically achieves an episode length of 7.

`Do graph of MCTS sims vs episode length with no training?`

Figure 1.7 shows the max-max nature of the tree search (after negating the action-value for the adversary). also highlights how Nsa is related to V_pred and U, with Nsa almost always being the right one.

1.1.2 Neural Network Training

c_{pareto} was set to 0.1. The minimum loss for the action value function is $-1bit \cdot c_{pareto}$. The neural networks get very close to this. Weirdly, the adversary losses are lower than the player losses on the first 2 policy iterations (which shouldn't happen as the adversary has no effect for these. After 10 iterations the action losses decrease to 0.1.

Figure 1.2: A zoomed in view of fig. 1.1. The three colours of each node are based on the value of $U = Q(\boldsymbol{x}, u) + ucb$ as in **??**, the number of state-action visits, $N(\boldsymbol{x}, u)$, and the state-cost, $c(\boldsymbol{x}_t)$.

arg1	arg2	arg3	arg4
param1	param2	param3	param4

Table 1.1: Training parameters used

After the second iteration, the action losses do not decrease much, suggesting that they have already been trained (possible over-fitting)

1.1.3 Action and State Values

show the evolution of action training, and how they start out biased due to the training with f2=0 and then gets less and less biased..

talk about general trends and what the optimal control policy is likely to be

how did things being discrete affect things? what might improve these results? Do average/smoothed one?

1.2 Performance

1.3 Performance Without an Adversary

1.4 Performance Against a Random Opponent

Selected Distributions

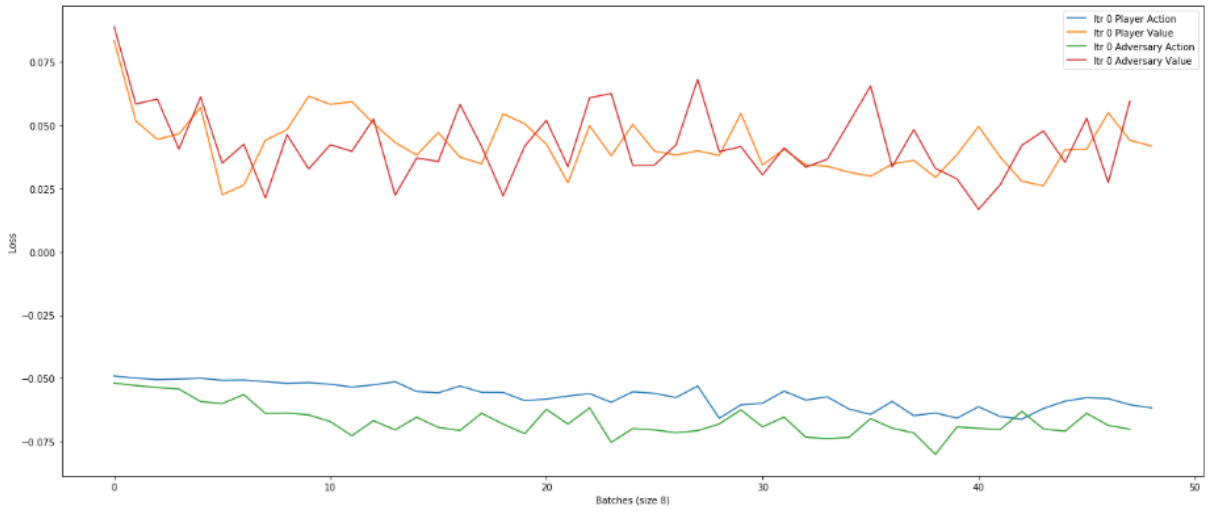

Figure 1.3: The losses for both the player and adversary neural networks after one policy iteration.

do against a constant random opponent do against say off for 30 steps then hit with a +- 5 do against a multinomial distribution with expected value of 0.05? do against

1.5 Performance Against a Trained Adversary

Compare when trained n the adversay against the random opponents? and also against no adversary? Is it robust?

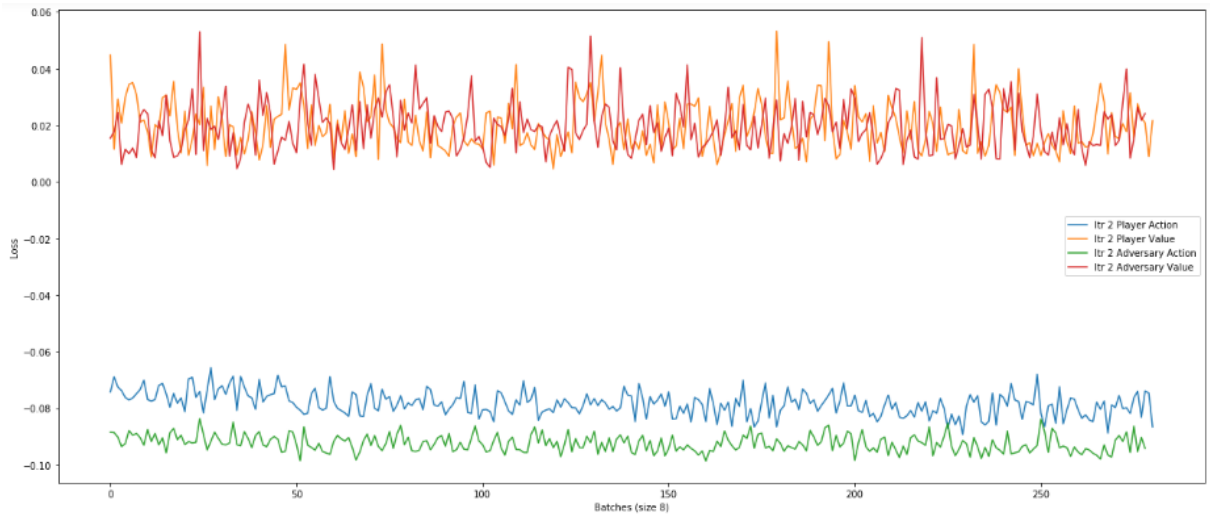

Figure 1.4: The losses for both the player and adversary neural networks after one policy iteration.

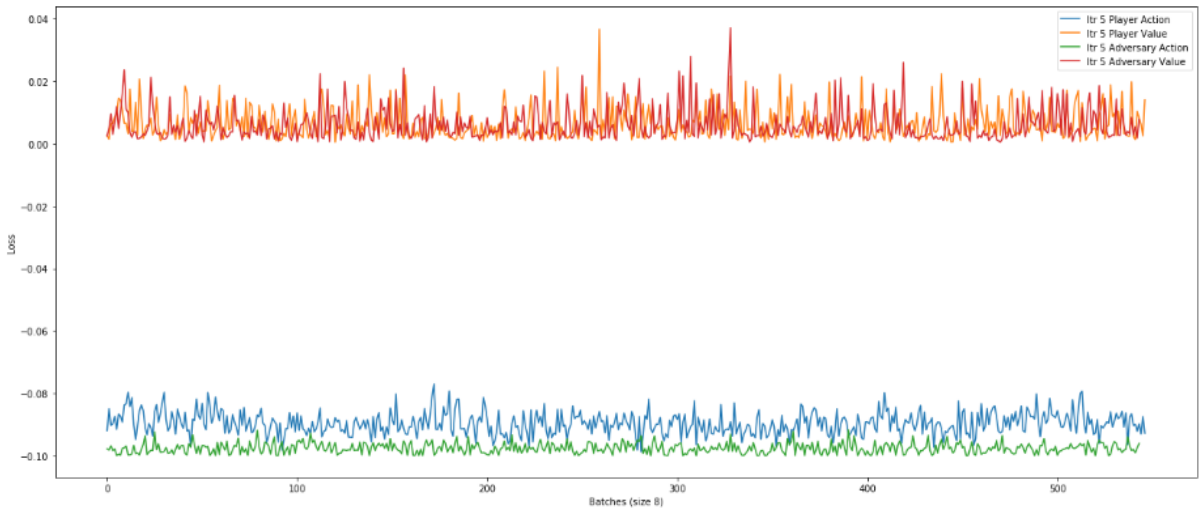

Figure 1.5: The losses for both the player and adversary neural networks after one policy iteration.

Figure 1.6: A zoomed in view of fig. 1.1. The three colours of each node are based on the value of $U = Q(\boldsymbol{x}, u) + ucb$ as in **??**, the number of state-action visits, $N(\boldsymbol{x}, u)$, and the state-cost, $c(\boldsymbol{x}_t)$.

Figure 1.7: A zoomed in view of fig. 1.1. The three colours of each node are based on the value of $U = Q(\boldsymbol{x}, u) + ucb$ as in **??**, the number of state-action visits, $N(\boldsymbol{x}, u)$, and the state-cost, $c(\boldsymbol{x}_t)$.