Enhancing the Exercise: From AlphaZero to MuZero in Traveling Salesperson

Proposal by Jonathn Schory and Jonathan Schwartz (js4979,js5115)

Abstract

The MuZero algorithm which combines a tree-based search (MCTS) with a learned neural model has proved to yield superhuman results in a variety of domains such as Chess, Go, Atari. Moreover, the MuZero algorithm does not require prior knowledge of domain dynamics, game rules, or pre-determined strategy. The course exercise, which implements AlphaX over an NP-Complete problem can therefore be enhanced by applying the MuZero methodology. With Traveling Salesperson Problem (TSP) as an evaluation domain at a manageable scale, we seek to demonstrate state-of-the-art performance with MuZero.

1. Problem, Goal, and Evaluation

The original exercise requires we implement AlphaX where X is any NP-complete problem over a graph. Choosing TSP as a NP-Complete domain, we seek to:

- Formulate the problem as a single-player game and use MCTS to solve a small random instance of the problem with 10 nodes. Actions in this case are limited to node manipulation (adding or removing nodes to the graph). User may be prompted to directly determine how and when actions are taken (limited to computationally feasible actions). Node features will be xy coordinates and graph edges represent distances.
- Add a Graph Neural Network representation to speed-up the MCTS, then implementing AlphaZero with self-play.
- The AlphaTSP will then be on a small random instance of the problem with 10 nodes. Then to be enhanced to MuZero TSP and evalated in a similar fashion on small scale dataset.
- Extend the dataset to a maximum of 20 nodes and thoroughly evaluate performance and limitations of MuZero implementation on an NP-Complete problem through plotting and probability analysis.

  A potential avenue for MuZero implementation evaluation on an agent is to implement it on a small-scale non-NP-Complete problem such as Connect4 provided in the AlphaZero repository.

2. Previous References

The key references in our paper is "Mastering Atari, Go, Chess and Shogi by planning with a learned model." This paper introduces the methodology, approach, evaluation, and implementation of MuZero in various domains. The analysis provided may aid in evaluating baseline performance and relative superiority of MuZero in novel domains.

3. Methodology and Algorithms

Inital algorithms and implementation for AlphaZero will make use of the AlphaZero General Github Repository. This will allow for foundation for the exercise as well as for MuZero enhancement. The approach will be utilize the strategy presented in the course exercise such that the progression is iterative. As such, the methodology would be to implement AlphaZero in a small scale non-NP domain, increase number of nodes, evaluate success, and repeat the process for TSP using a successfully implemented MuZero framework.

4. Next Steps

Potential following steps for our project include but are not limited to:

- Scaling MuZero to larger dataset and evaluate performance on graphs that exceed 20 nodes.
- Evaluate performance on variety of NP-Complete problems which may include: vertex-cover problem, Hamiltonian-cycle problem, set-covering prob-

lem, independent-set problem, graph-coloring problem, clique problem, or longest-simple-cycle.

- Experiment with variety of neural architectures for MuZero and/or introduce model ensembling to reach top performance.

5. References

[1] Schrittwieser, Julian, et al. "Mastering atari, go, chess and shogi by planning with a learned model." arXiv preprint arXiv:1911.08265 (2019).