A Puzzled AI; Challenging a Computer to Solve Mazes

Reese Allen, Igancio Segovia-Dominguez
WVU School of Mathematical and Data Sciences

West Virginia University
BENJAMIN M. STATLER COLLEGE OF ENGINEERING AND MINERAL RESOURCES

West Virginia University
EBERLY COLLEGE OF ARTS AND SCIENCES

Problem Statement

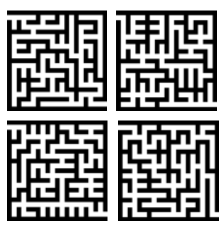
Create a reinforcement learning model that incorporates Monte Carlo methods and can be used to solve mazes

Background

- Reinforcement learning is a widely used machine learning method
- Applications in robotics, transportation, and natural language processing
- Primary components used in episodic evolution:
  - A given **state** of environment
  - A list of possible **actions**
  - A **policy** to connect actions to their **rewards**
- Monte Carlo methods
  - Average returns from sampled sequences
  - Do not require the state to select actions

Dataset

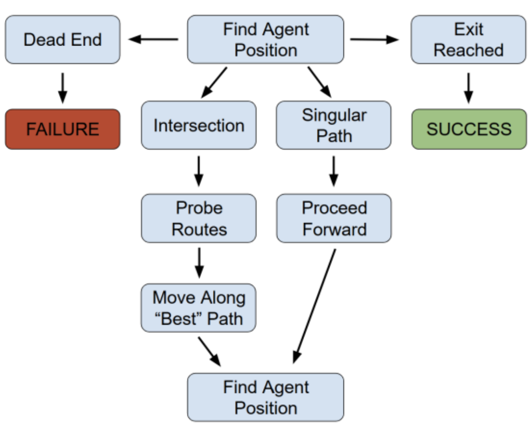
- Mazes were taken from an open-source pool of randomly generated mazes
- Maze imaging was converted into arrays of 21 x 21 pixels
- 1000 mazes used to test algorithm
- All mazes maintained consistent entrance and exit coordinates
- All puzzles were solvable



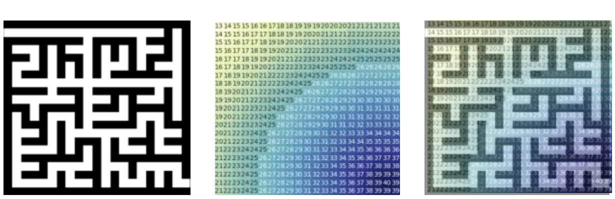
Sample mazes from Kaggle dataset

Reinforcement Learning Model

- Model state determined by position and available paths
- Monte Carlo method of path simulation used at intersections
- Rewards determined by proximity grid and path course
- Path of highest reward selected as the "best" path
- Algorithm proceeds until either a dead end or the exit has been reached



Algorithm decision process for maze navigation



Demonstration of how proximity weights correlate to positions within the maze

Results

- Final model solved **59.0%** of the 1000 mazes (using wandering paths of up to 25 steps)
- Algorithm solved all mazes in **1:16.34** on personal laptop

Conclusions

- Monte Carlo method of averaging proved less effective than pursuing maximum reward
  - Accuracy of 78.6% to 38.6% for maximum seeking and Monte Carlo respectively (15 wandering steps)
  - By the nature of means and random pathing at every intersection, dead ends near intersections cause confounding obstacles
  - Increasing the number of wandering steps allows more successful actions to create a larger impact on the average reward
- Maze solution success directly proportional to the number of wandering steps
- Significantly lower and significantly higher position weight scaling yields lower success than a scale of 40

Future Work

- Expand the concept of traversing a reward distribution to more complicated mazes (e.g. circular mazes, non-standard geometry mazes, mazes with clearings, etc.)
- Generalize the solution method to solve similar problems in other field/puzzles/games
- Test Monte Carlo methods and maximum pursuit against deep learning for the same problem

References

GitHub Repository

LinkedIn Page