Path Finding Project

Evolutionary vs Q-Learning

Nicholas Fantino-Dyer
5/3/24

Purpose of the Study:

To compare the effectiveness of different machine learning models in learning how to navigate a maze.

The models evaluated are:

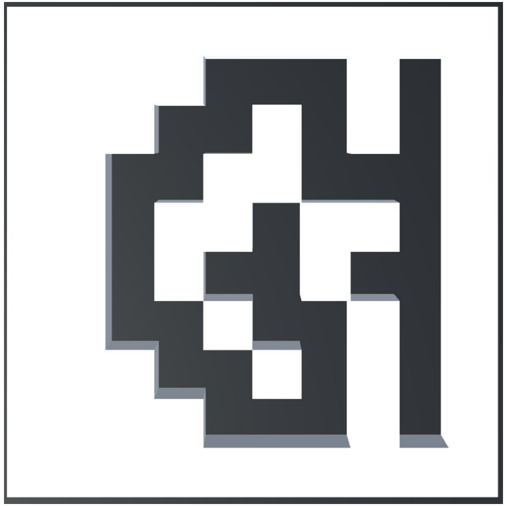
- Evolutionary Model
- Q-Learning
- A* Algorithm.

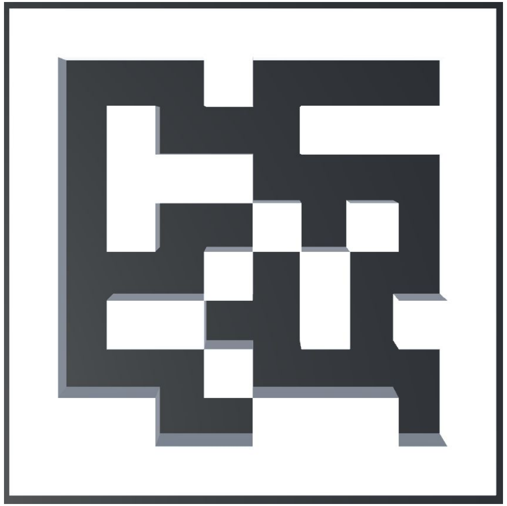
Overview of Models

- Evolutionary Model:
  - Mimics natural selection by evolving creatures over generations, using mechanisms like mutation and fitness-based reproduction.
- Q-Learning:
  - A form of reinforcement learning that updates the value of actions in each state based on rewards, which aims to discover the optimal strategy through trial and error.
- A* Algorithm:
  - A* pathfinding uses heuristics to estimate distances to the goal, and branches off the nodes with the lowest heuristic.

Maze Environment and Design

The maze is a randomly generating 10 by 10 grid. This allows for each run to be different to be able to accurately test how well these models learn in different environments.

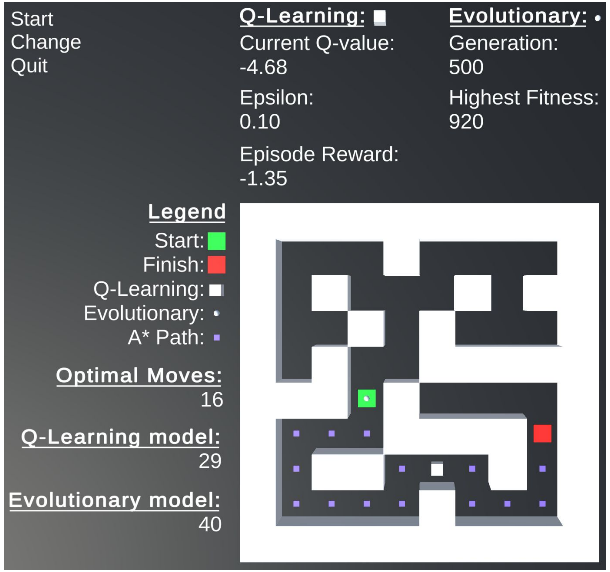


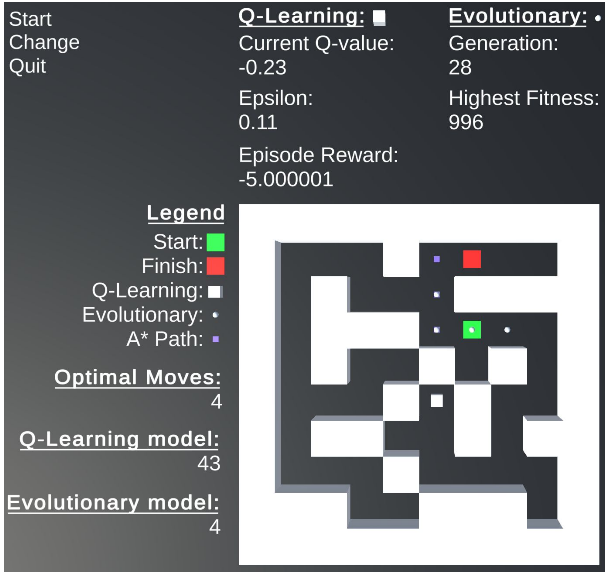


Models in Action

In this example, the Q-Learning model is still in the predominantly random movement phase, and on average takes 43 moves to reach the goal. However, the Evolutionary Model has reached the goal in the optimal moves. This is due to the goal being in such close proximity to the start. The evolutionary model flourishes when there isn't a lot of distance as the more moves it takes to reach the end, the more genes are required to be correct.

In this example, Evolutionary model has completed 500 generations and reached the goal after 40 moves. However, the Q-Learning model at this point has learned not to hit as many walls and has an average of 29 moves to reach the goal. The reason the Evolutionary performed worse was because the reaching the goal without any errors, required there to be a string of 16 genes in a row with the correct order of movements.





Detail Analysis

Evolutionary Model

Genes: Each creature was randomly assigned 50 genes one for each move (left, right, up, down).

Selection Process: After a creature either reaches the goal or exhausts its genes, it is assigned a fitness. Then sorting from highest to lowest fitness in an array of creatures, the highest two are selected for the new generation to copy their genes with a mutation rate of 10%. If the creature reaches the goal before all the genes are used, the remaining genes are removed as they are no longer necessary.

Fitness: If the creature moves into a wall then -10 fitness, if it doesn't reach the goal in its run -100 fitness, and each movement cost -1 fitness as well. Then it determines how close it was to the goal and assigns fitness accordingly. However if it reaches the goal then it gets +1000 fitness.

Q-Learning Model

Rewards and Penalties: Each time the model steps, it's given a -0.05 reward to encourage efficiency, and is given +1 for reaching the goal.

Reset: After the model would finish its 50 steps, or reached the goal, the reward would be given and it would store that episode's data in the q-table.

A* Algorithm

Optimal Path: The optimal path is found by creating nodes on the surrounding open blocks from the starting node. From there it finds which one is closest to the goal and expands that way, getting neighbors of each node. If that way is blocked off then it starts expanding from the next closest node. When it reaches the goal, it gets the ancestor nodes of that path so that it can trace its way back to the start.

Comparative Analysis

Evolutionary Model

Strengths:

- Can evaluate multiple solutions simultaneously, which allows this model to train much faster
- Once an efficient path is found, mutation can still occur, which allows for some adaptation if the environment changes.

Weaknesses:

- Performance heavily depends on initial genes and random mutations, which leads to inconsistent effectiveness across runs.
- If the fitness calculation is not good enough, then the model can fail to complete the maze.

Q-Learning Model

Strengths:

- Once sufficient learning has occurred, Q-Learning can be very effective at finding the optimal path, even in changing environments.

Weaknesses:

- Learning the optimal path can be slow, which requires many iterations to explore and update the Q-Table.

A* Algorithm

Strengths:

- Highly efficient in finding the shortest path due to its heuristic nature, which guides the search.

Weaknesses:

- Requires significant memory to store all nodes in the pathfinding process, which can be a constraint in very large mazes.

Conclusion

Summary of Findings:

Evolutionary Model:

Effectiveness: This model proved most effective in mazes where there was not a complex path, mainly due to the way I calculated fitness.

Optimal Use Case: Best use case is when the goal is straight ahead or close by the start.

Q-Learning:

Effectiveness: Q-Learning was most effective in environments that allowed for extensive trial and error, which enabled the model to outperform the evolutionary model.

Optimal Use Case: Any maze but given enough time to train.

A* Algorithm:

Effectiveness: A* served its purpose of showing the user an optimal path to the goal. It was especially effective in this smaller maze.

Optimal Use Case: A maze smaller than 1000x1000.

Future

Some improvements to make to the Evolutionary Model is the way I calculated fitness. If I had it use the A* path as a guide instead of just the distance from the goal, it would perform much better.

Some improvements for the Q-Learning model would be adding some hazards or items to pick up on the way to the goal, so that it can be fully utilized. Just having a start and end for this model doesn’t outweigh the time it takes to train it.