Self-Driving Car project with Q-learning and NEAT

GROUP MEMBERS

AKSHAT KUMAR
ADITYA PRAKASH

INTRODUCTION

In this project, we implemented an AI car simulation using Q-learning, a type of reinforcement learning, in conjunction with the NEAT (NeuroEvolution of Augmenting Topologies) algorithm. The objective was to train AI cars to navigate a track while avoiding collisions using Q-learning to learn optimal policies.

Q-LEARNING

Q-learning is a model-free reinforcement learning algorithm that learns an optimal policy by iteratively updating Q-values for state-action pairs. In our simulation, each state corresponds to a configuration of the car on the track, and each action represents a possible steering or speed adjustment.

NEAT ALGORITHM

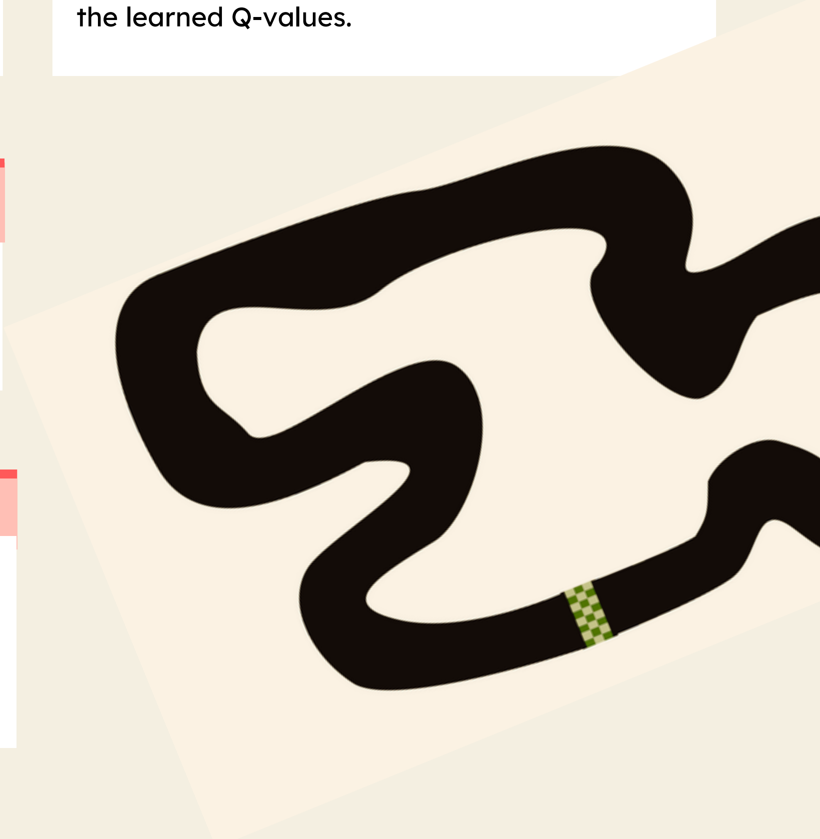
The NEAT algorithm was used to evolve neural networks that control the AI cars. Each neural network receives inputs representing the car's environment (e.g., sensor readings) and outputs actions (e.g., steering and speed adjustments) based on the learned Q-values.

BELLMAN EQUATION:

$$Q(S,A)=Q(S,A)+A\cdot(R+\Gamma\cdot MAXQ(S',A')-Q(S,A))$$

IMPLEMENTATION

WE IMPLEMENTED THE SIMULATION USING PYTHON WITH THE PYGAME LIBRARY. EACH AI CAR WAS REPRESENTED BY A SPRITE THAT COULD MOVE AND ROTATE ON THE TRACK. THE TRACK LAYOUT WAS DEFINED USING A MAP IMAGE, WITH BORDERS REPRESENTED BY A SPECIFIC COLOR.



TRAINING AND EVALUATION

TRAINING
DURING TRAINING, THE NEAT ALGORITHM EVOLVED NEURAL NETWORKS FOR THE AI CARS OVER MULTIPLE GENERATIONS. EACH GENERATION CONSISTED OF A POPULATION OF CARS, WITH THE FITTEST CARS SELECTED FOR BREEDING BASED ON THEIR PERFORMANCE IN THE SIMULATION.

EVALUATION
THE FITNESS OF EACH CAR WAS DETERMINED BY THE DISTANCE TRAVELED BEFORE CRASHING OR COMPLETING A LAP. THE NEURAL NETWORKS WERE EVALUATED BASED ON THEIR ABILITY TO PRODUCE CARS THAT COULD NAVIGATE THE TRACK SUCCESSFULLY USING THE LEARNED Q-VALUES.

RESULTS

After several generations of evolution, the AI cars showed significant improvement in their ability to navigate the track and avoid collisions. They learned optimal policies for steering and speed adjustment, demonstrating the effectiveness of Q-learning in training AI agents for complex tasks.

CONCLUSION

In conclusion, we successfully developed an AI car simulation using Q-learning with the NEAT algorithm. The project highlights the potential of combining reinforcement learning techniques with evolutionary algorithms to train AI agents for tasks such as autonomous driving.

FUTURE WORK

FUTURE WORK COULD INVOLVE FURTHER OPTIMIZING THE Q-LEARNING PARAMETERS AND NEURAL NETWORK ARCHITECTURE TO IMPROVE THE PERFORMANCE OF THE AI CARS. ADDITIONALLY, EXPANDING THE SIMULATION TO INCLUDE MORE COMPLEX TRACK LAYOUTS AND OBSTACLES COULD PROVIDE A MORE CHALLENGING ENVIRONMENT FOR TRAINING AND TESTING AI AGENTS.